Class Example 1 – Linear Regression Example

The code below uses "BrisGCtemp.csv", as appeared in a class example of Assignment 3. This file contains temperature observations recorded in Brisbane and the Gold Coast.
We are looking for a model of the form:

$$\text{Gold Coast Temperature} = \beta_0 + \beta_1 \text{Brisbane Temperature} + \epsilon.$$

We slice up the data over different months, yielding a different model for each month. That is, each month has it's own β_0 and β_1.

As presented below, the code is for the month of May.

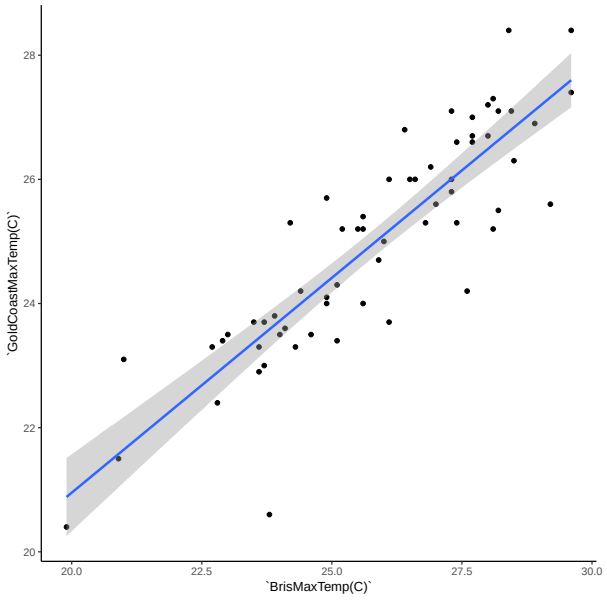
```
> library(tidyverse)
> temps <- read_csv("BrisGCtemp.csv")
> # This is the desired month to filter by
> desiredMonth <- 5
>
> print(temps)
> # column 2 in the dataframe is the month, filter by it
> tempFilter <- filter(temps, Month==desiredMonth)
> nrow(tempFilter)
>
> model <- lm(`GoldCoastMaxTemp(C)`~`BrisMaxTemp(C)`,data=tempFilter)
>
> paste("For Month", desiredMonth, "observations", nrow(tempFilter),
+ "means (",mean(round(tempFilter$`BrisMaxTemp(C)`),4),
+ round(mean(tempFilter$`GoldCoastMaxTemp(C)`),4),")")
>
> ggplot(tempFilter,aes(x=`BrisMaxTemp(C)`,y=`GoldCoastMaxTemp(C)`))+
+ geom_point() + geom_smooth(method="lm")+theme_classic()
>
> summary(model)
```

The output present on the next page shows:

- A plot of the data and the regression line.
- A shortened version of the dataframe (table).
- Summary means (Brisbane,Gold Coast) and observation counts for the selected month (May in this case).
- The estimated model. In this case,

$$\text{Gold Coast Temperature} = 7.10 + 0.6922 \times \text{Brisbane Temperature}.$$

```
# A tibble: 777 x 5
    Year Month   Day `BrisMaxTemp(C)` `GoldCoastMaxTemp(C)`
   <dbl> <dbl> <dbl>            <dbl>                  <dbl>
 1  2017     2    15             30.4                   30.4
 2  2017     2    14             30.3                   29.9
 3  2017     2    13             35.6                   31.5
 4  2017     2    12             37.6                   33
 5  2017     2    11             37                     32.6
 6  2017     2    10             33                     29
 7  2017     2     9             32.2                   30.3
 8  2017     2     8             32.8                   31
 9  2017     2     7             32.8                   30.5
10  2017     2     6             33                     30.5
# ... with 767 more rows
[1] "For Month 5 observations 62 means ( 26 24.971 )"

Call:
lm(formula = `GoldCoastMaxTemp(C)` ~ `BrisMaxTemp(C)`, data = tempFilter)

Residuals:
     Min       1Q   Median       3Q      Max
-2.98256 -0.45192  0.07139  0.47908  1.63309

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)        7.1070     1.2923   5.499 8.27e-07 ***
`BrisMaxTemp(C)`   0.6923     0.0499  13.873  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8652 on 60 degrees of freedom
Multiple R-squared:  0.7623,Adjusted R-squared:  0.7584
F-statistic: 192.5 on 1 and 60 DF,  p-value: < 2.2e-16
```



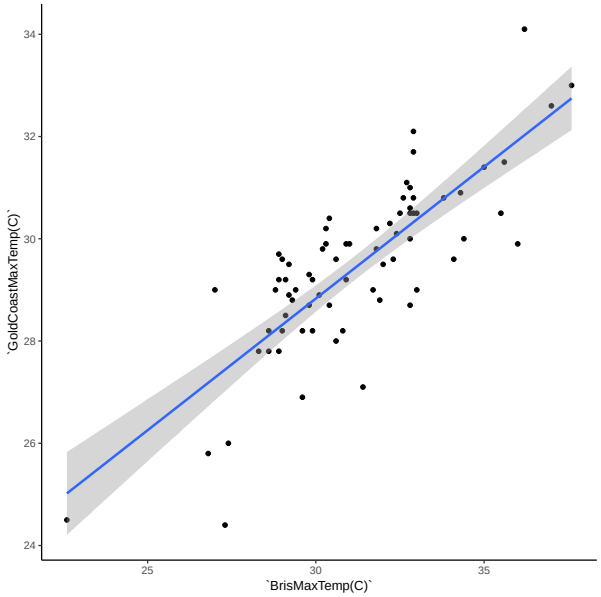
(a) Run May and understand the result.

Solution: Looking above we can see that there are 62 observations for May, the mean maximum temperature for Brisbane was 26 °C and for the Gold Coast was 24.971 °C. Looking at the linear model there is a strong relationship between the two temperatures.

(b) Modify the code (`desiredMonth`) for a different month and analyse the results.

Solution: For example for February we have,

```
> # This is the desired month to filter by
> desiredMonth <- 2
>
> # column 2 in the dataframe is the month, filter by it
> tempFilter <- filter(temps, Month==desiredMonth)
>
> model <- lm(`GoldCoastMaxTemp(C)`~`BrisMaxTemp(C)`,data=tempFilter)
>
> paste("For Month", desiredMonth, "observations", nrow(tempFilter), "means (",mean(rou

[1] "For Month 2 observations 72 means ( 31 29.4528 )"

> ggplot(tempFilter,aes(x=`BrisMaxTemp(C)`,y=`GoldCoastMaxTemp(C)`))+geom_point() + geo
>
> summary(model)


Call:
lm(formula = `GoldCoastMaxTemp(C)` ~ `BrisMaxTemp(C)`, data = tempFilter)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0400 -0.4347  0.1069  0.5804  2.0749

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      13.37539    1.41394    9.46 3.82e-14 ***
`BrisMaxTemp(C)`  0.51519    0.04515   11.41  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9981 on 70 degrees of freedom
Multiple R-squared:  0.6503,Adjusted R-squared:  0.6453
F-statistic: 130.2 on 1 and 70 DF,  p-value: < 2.2e-16
```



Looking at this we see there are now 72 observations and the mean of Brisbane's Maximum Temperature is 31 °C and for the Gold Coast is it 29.45 °C. The model is again significant with a positive correlation between the two areas. The R^2 value is lower so the model explains less of the variation in the Gold Coast Maximum Temperature.

(c) Assume you accept this model and observe a temperature of 30 °C in Brisbane in a given day. What is your estimate of the temperature in the Gold Coast for that day? (i) In January? (ii) In October?

Solution:

(i) First we need to calculate the model for January which we do with

```
> # This is the desired month to filter by
> desiredMonth <- 1
>
> # column 2 in the dataframe is the month, filter by it
> tempFilter <- filter(temps, Month==desiredMonth)
>
> model <- lm(`GoldCoastMaxTemp(C)`~`BrisMaxTemp(C)`,data=tempFilter)
> C <- coef(model)
> print(C)

     (Intercept) `BrisMaxTemp(C)`
      11.6161621        0.5836163
```

Now that we have the slope and intercept values we can calculate the predicted temperature when Brisbane is at 30 °C.

$$\text{Gold Coast Max Temperature} = 11.616 + 0.584 \times 30 = 29.1246$$

Or in R

```
> predict(model,new=tibble(`BrisMaxTemp(C)`=30),type="response")

       1
29.12465
```

(ii) For October we can do the same procedure

```
> # This is the desired month to filter by
> desiredMonth <- 10
>
> # column 2 in the dataframe is the month, filter by it
> tempFilter <- filter(temps, Month==desiredMonth)
>
> model <- lm(`GoldCoastMaxTemp(C)`~`BrisMaxTemp(C)`,data=tempFilter)
> C <- coef(model)
> print(C)

     (Intercept) `BrisMaxTemp(C)`
       11.793261         0.518582

> predict(model,new=tibble(`BrisMaxTemp(C)`=30),type="response")

       1
27.35072
```

Class Example 2 – Automation on All Months

The code below obtains a regression model for all 12 months.

```
> library(tidyverse)
> temps <- read_csv("BrisGCtemp.csv")
> # Create a data frame of summary statistics
> monTemps <- temps %>% group_by(Month) %>% summarise(NoPoints = n(),
+ `BrisMeanMaxTemp(C)`=mean(`BrisMaxTemp(C)`),
+ `GoldCoastMeanMaxTemp(C)`=mean(`GoldCoastMaxTemp(C)`))
>
> # Create an empty data frame to store results
> models<-tibble("Intercept"=double(),"Slope"=double())
>
> # For loop to create all the linear models
> for(desiredMonth in 1:12) {
+         monthData <- filter(temps,Month == desiredMonth)
+         model <- lm(`GoldCoastMaxTemp(C)`~`BrisMaxTemp(C)`,data=monthData)
+         models[desiredMonth,1:2] <- coef(model)
+ }
>
> # Combine all the columns together
> monTemps <- bind_cols(monTemps,models)
> print(monTemps)
```

(a) Run the code above and interpret the results.

Solution: Running the code above we get the following output:

```
# A tibble: 12 x 6
   Month NoPoints `BrisMeanMaxTemp(C~ `GoldCoastMeanMaxTem~ Intercept Slope
   <dbl>    <int>               <dbl>                 <dbl>     <dbl> <dbl>
 1     1       93                30.8                  29.6     11.6  0.584
 2     2       72                31.2                  29.5     13.4  0.515
 3     3       62                30.0                  29.0     12.9  0.534
 4     4       60                27.5                  27.0      9.84 0.624
 5     5       62                25.8                  25.0      7.11 0.692
 6     6       60                21.6                  21.4      5.46 0.741
 7     7       62                21.8                  21.4      3.85 0.804
 8     8       62                23.4                  22.6      5.45 0.731
 9     9       60                24.8                  23.9      5.87 0.729
10    10       62                27.1                  25.8     11.8  0.519
11    11       60                29.7                  28.0     11.9  0.543
12    12       62                29.7                  28.5      9.04 0.658
```

Looking at this output we can see that most months have two years of data, except January which has 3 years and February which has about 2 and a half years. We can also see that the temperatures in the middle of the year are lower than those at the beginning and end of the year. All the coefficients are positive which means there is a positive correlation between the Gold Coast and Brisbane temperatures. The slope gets steeper during the winter months but the intercept goes lower.

(b) Add a column of P-values for the slope coefficient using "`summary(model)$coefficients[2,3:4]`" and re-run to obtain the values for all 12 months. Assume that for every month you are considering $H_0 : \beta_1 = 0$ vs. $H_1 : \beta_1 \neq 0$. What are your results for these 12 hypothesis tests?

Solution: Creating these columns we use the following code,

```
> # Summary statistics
> monTemps <- temps %>% group_by(Month) %>% summarise(NoPoints = n(),
+ `BrisMeanMaxTemp(C)` <- mean(`BrisMaxTemp(C)`),
+ `GoldCoastMeanMaxTemp(C)` <- mean(`GoldCoastMaxTemp(C)`))
>
> # empty dataframe for output from linear models
> models <- tibble("Intercept"=double(),"Slope"=double(),"T-value"=double(),
+ "P-Value"=double())
>
> # creating all the linear models
> for(desiredMonth in 1:12) {
+         monthData <- filter(temps,Month == desiredMonth)
+         model <- lm(`GoldCoastMaxTemp(C)`~`BrisMaxTemp(C)`,data=monthData)
+         models[desiredMonth,1:2] <- coef(model)
+         models[desiredMonth,3:4] <- summary(model)$coefficients[2,3:4]
+ }
>
> monTemps <- bind_cols(monTemps,models)
> # Printing selected columns
> print(monTemps[,c(1:2,5:8)])
```

```
# A tibble: 12 x 6
   Month NoPoints Intercept Slope `T-value` `P-Value`
   <dbl>    <int>     <dbl> <dbl>     <dbl>     <dbl>
 1     1       93     11.6  0.584     12.4   2.69e-21
 2     2       72     13.4  0.515     11.4   1.25e-17
 3     3       62     12.9  0.534      2.69  9.31e- 3
 4     4       60      9.84 0.624     11.4   1.66e-16
 5     5       62      7.11 0.692     13.9   2.22e-20
 6     6       60      5.46 0.741     12.3   9.93e-18
 7     7       62      3.85 0.804     18.7   1.06e-26
 8     8       62      5.45 0.731     10.9   7.26e-16
 9     9       60      5.87 0.729     10.6   3.02e-15
10    10       62     11.8  0.519      8.22  2.10e-11
11    11       60     11.9  0.543      9.20  6.14e-13
12    12       62      9.04 0.658     10.4   5.13e-15
```

Looking at the twelve hypothesis tests we can see that all months have strong evidence to support that the coefficients are significantly different from zero. The lowest P-value is in March, but still is quite strong evidence.

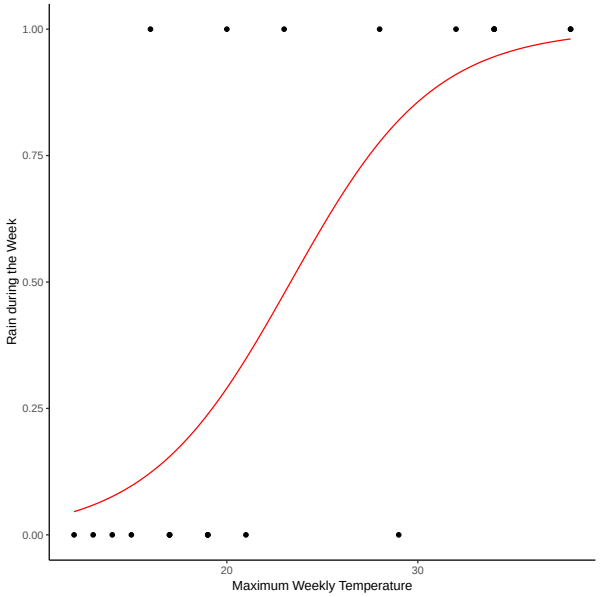
Class Example 3 – Logistic Regression

Logistic regression is a mechanism to describe "0"/"1" outcomes (y) as a function of (x). Here we are predicting the probability of an event happening ($y = 1$) as a function of x.

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x.$$

The following code takes maximal weekly temperatures (`x`) and the event of having rain during that week (`y`). It then constructs a logistic regression model where C are the resulting coefficients.

```
> x <- c(23,32,17,34,29,38,16,14,19,34,38,19,20,21,19,17,13,12,28,34,15,17)
> y <- c(1,1,0,1,0,1,1,0,0,1,1,0,1,0,0,0,0,0,1,1,0,0)
> logdata <- tibble(x,y)
>
> logmodel <- glm(y~x,data=logdata,family="binomial")
>
> C <- coef(logmodel)
>
> xm <- seq(min(x),max(x),length=100)
> ym <- 1/(1+exp(-(C[1]+C[2]*xm)))
> xlim <- c(min(x),max(x))
> ggplot() + geom_point(aes(x=x,y=y)) + geom_line(aes(x=xm,y=ym),colour="red") +
+  labs(x="Maximum Weekly Temperature",y="Rain during the Week") + theme_classic()
>
> summary(logmodel)


Call:
glm(formula = y ~ x, family = "binomial", data = logdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8516  -0.5798  -0.3275   0.3346   2.0478

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -6.2505     2.3420  -2.669  0.00761 **
x             0.2678     0.1052   2.546  0.01091 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 30.316  on 21  degrees of freedom
Residual deviance: 16.757  on 20  degrees of freedom
AIC: 20.757

Number of Fisher Scoring iterations: 5
```



(a) Assume the temperature is 26 °C, what is your estimate for the probability of having rain?

Solution:
To calculate the probability of having rain, first we need to obtain the odds of having rain during the week with a maximal temperature of 26 °C. We do this by putting the new value into the equation to obtain the log odds of rain.

$$\begin{aligned}\log(\text{Odds of Rain}) &= -6.2504936 + 0.2677972 \times 26 \\ &= 0.7122333\end{aligned}$$

Now taking the exponetial of each side.

$$\begin{aligned}\text{Odds of Rain} &= e^{0.7122333} \\ &= 2.0385388\end{aligned}$$

Now convert the odds to a probability

$$\begin{aligned}P(Rain) &= \frac{1}{1 + \frac{1}{\text{Odds of Rain}}} \\ &= 0.6708944\end{aligned}$$

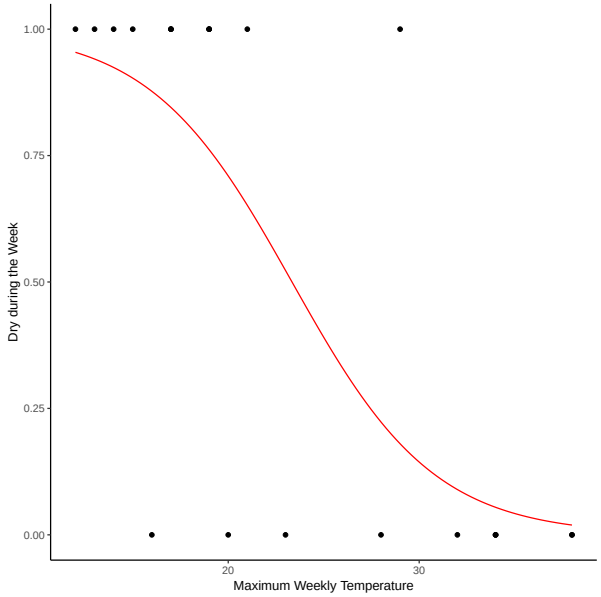
This can also be calculated in R with the following command

```
> predict(logmodel,new=tibble(x=26),type="response")

        1
0.6708944
```

(b) Modify the code by adding a line "`y<-1-y`" after setting up `y`. Explain the output.

Solution:
Modifying the code bu adding the line request we get the following.

```
> x <- c(23,32,17,34,29,38,16,14,19,34,38,19,20,21,19,17,13,12,28,34,15,17)
> y <- c(1,1,0,1,0,1,1,0,0,1,1,0,1,0,0,0,0,0,1,1,0,0)
> y<-1-y
> logdata <- tibble(x,y)
>
> logmodel <- glm(y~x,data=logdata,family="binomial")
>
> C <- coef(logmodel)
>
> xm <- seq(min(x),max(x),length=100)
> ym <- 1/(1+exp(-(C[1]+C[2]*xm)))
> xlim <- c(min(x),max(x))
> ggplot() + geom_point(aes(x=x,y=y)) + geom_line(aes(x=xm,y=ym),colour="red") +
+ labs(x="Maximum Weekly Temperature",y="Dry during the Week") + theme_classic()
>
> summary(logmodel)


Call:
glm(formula = y ~ x, family = "binomial", data = logdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0478  -0.3346   0.3275   0.5798   1.8516

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   6.2505     2.3420   2.669  0.00761 **
x            -0.2678     0.1052  -2.546  0.01091 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 30.316  on 21  degrees of freedom
Residual deviance: 16.757  on 20  degrees of freedom
AIC: 20.757

Number of Fisher Scoring iterations: 5
```



Since y was originally whether it rained or not, this new model predicts the likelihood of a week being dry given the weekly maximal temperature.