Stats_Term_Project

Rachael Joan Dias

22/11/2018

```
library(ggplot2)
library(Stat2Data)
library(broom)
library(dplyr)
```

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

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

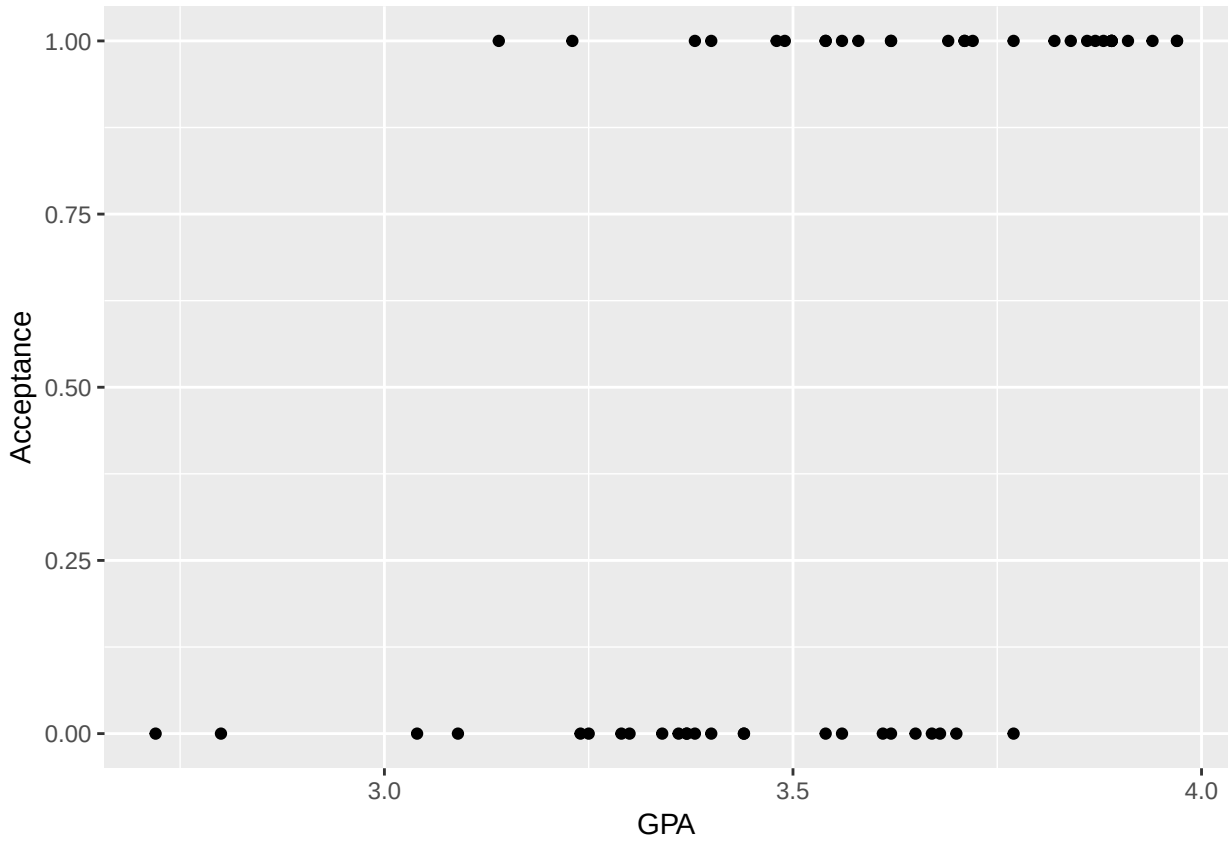
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#Load the MedGPA data from the Stats2Data library
data("MedGPA")

#Select the GPA and the Acceptance
head(MedGPA%>%select(GPA,Acceptance))
```

```
##    GPA Acceptance
## 1 3.62          0
## 2 3.84          1
## 3 3.23          1
## 4 3.69          1
## 5 3.38          1
## 6 3.72          1
```

```
#Visualize the data using a scatter plot
data_points<-ggplot(data = MedGPA, aes(y = Acceptance, x = GPA))+geom_point()
data_points
```

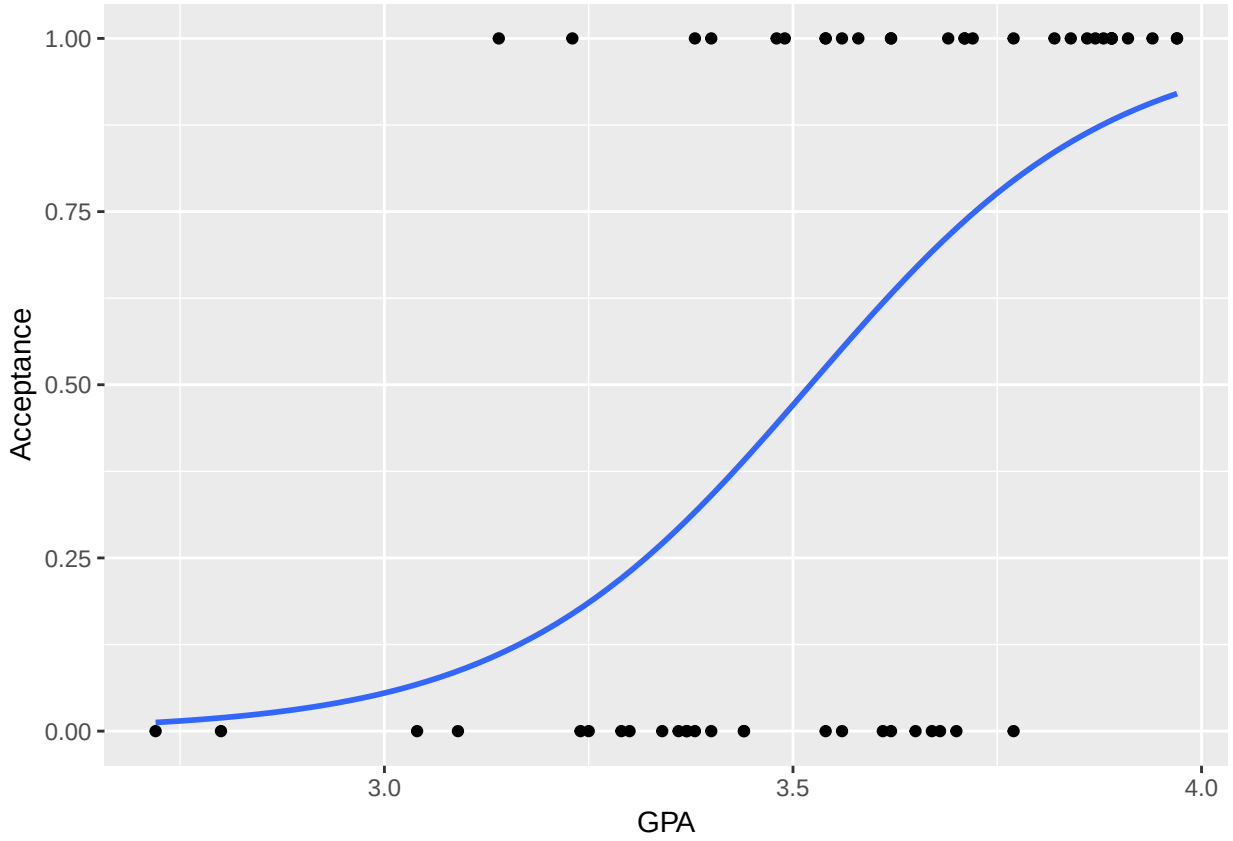



```
#Fit the logistic model using the glm function
mod<-glm(Acceptance~GPA, data =MedGPA, family = binomial)
summary(mod)
```

```
##
## Call:
## glm(formula = Acceptance ~ GPA, family = binomial, data = MedGPA)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7805  -0.8522   0.4407   0.7819   2.0967
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -19.207      5.629  -3.412 0.000644 ***
## GPA            5.454      1.579   3.454 0.000553 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 75.791  on 54  degrees of freedom
## Residual deviance: 56.839  on 53  degrees of freedom
## AIC: 60.839
##
## Number of Fisher Scoring iterations: 4
```

```r
#Visualize the data with the logistics curve
data_point<-ggplot(data = MedGPA, aes(y = Acceptance, x = GPA))+geom_point()

#Add the logistics curve
data_point + geom_smooth(method = "glm", se = FALSE, method.args = list(family = "binomial"))
```




```r
#Bin the GPA column and find the average GPA and Acceptance Rate for each bin
MedGPA_binned<-MedGPA%>%mutate(Bins = cut(GPA, breaks=seq(0, 4, by=.20)))%>%group_by(Bins)%>%summarize(m
MedGPA_binned
```

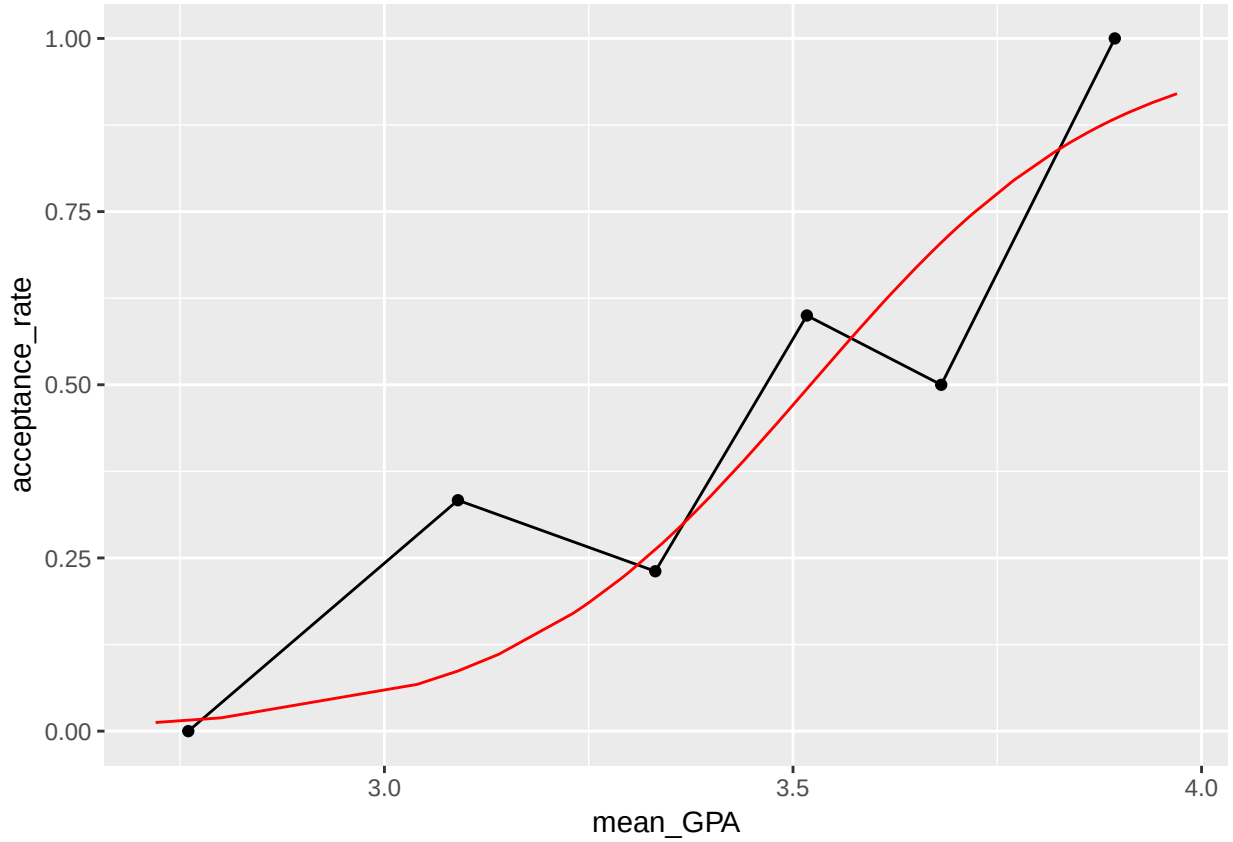
```
## # A tibble: 6 x 3
##   Bins      mean_GPA acceptance_rate
##   <fct>        <dbl>           <dbl>
## 1 (2.6,2.8]     2.76           0
## 2 (3,3.2]       3.09           0.333
## 3 (3.2,3.4]     3.33           0.231
## 4 (3.4,3.6]     3.52           0.6
## 5 (3.6,3.8]     3.68           0.5
## 6 (3.8,4]       3.89           1
```

```r
#Plot the binned Mean GPA
data_point<-ggplot(data = MedGPA_binned, aes(x = mean_GPA, y = acceptance_rate)) +
  geom_point() + geom_line()

#Augment the model, type.predict argument has been set to "response" to ensure fitted values are on the
MedGPA_plus<-mod%>%augment(type.predict = "response")

#plot logistic model on probability scale
```

```
data_point +geom_line(data = MedGPA_plus, aes(x = GPA, y = .fitted), color = "red")
```

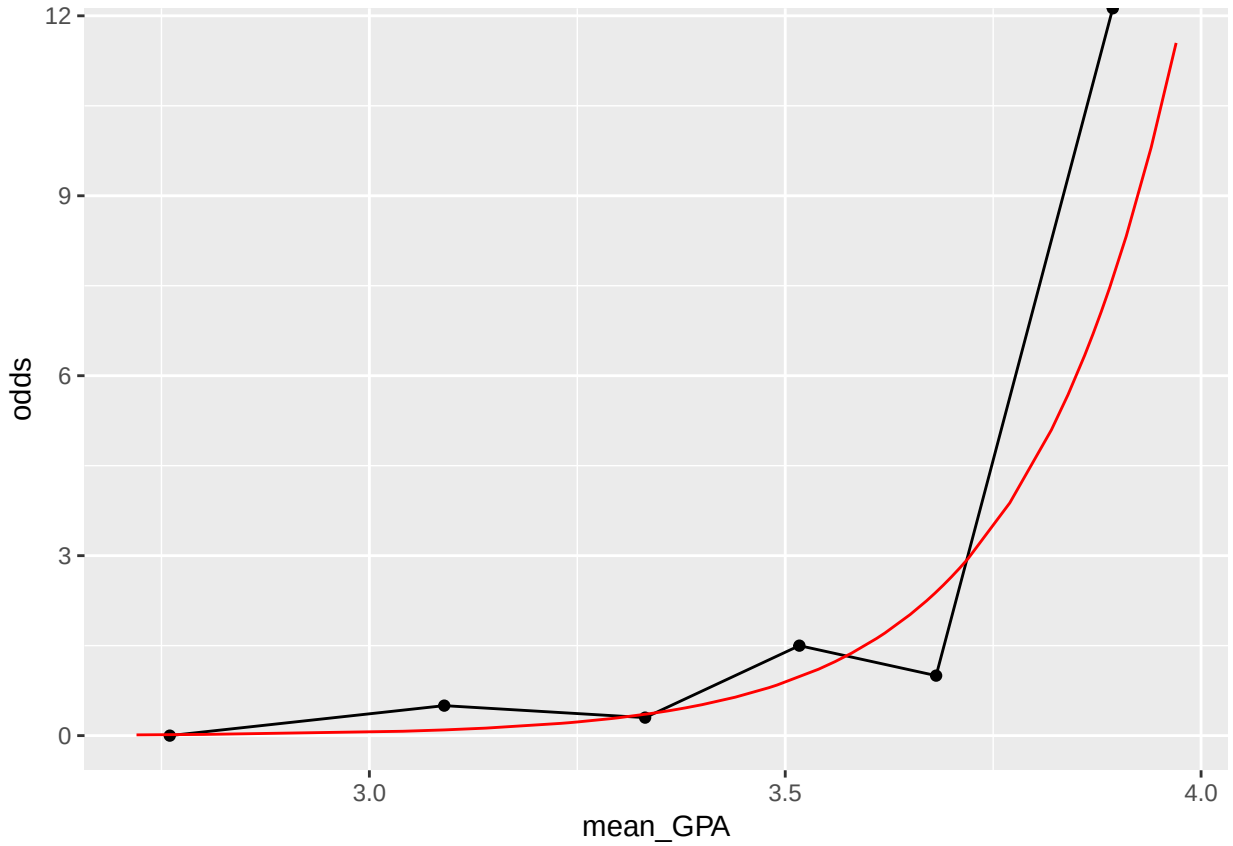



```
#Add Odds column to find the odds of acceptance
MedGPA_binned<-MedGPA_binned %>%mutate(odds = acceptance_rate / (1 - acceptance_rate))

#Plot the binned odds
data_point<-ggplot(data = MedGPA_binned, aes(x = mean_GPA, y = odds)) + geom_point() +geom_line()

#compute odds for observations
MedGPA_plus<-MedGPA_plus %>%mutate(odds_hat = .fitted / (1 - .fitted))

#Plot the logistic model on the odds scale
data_point+geom_line(data = MedGPA_plus, aes(x = GPA, y = odds_hat), color = "red")
```

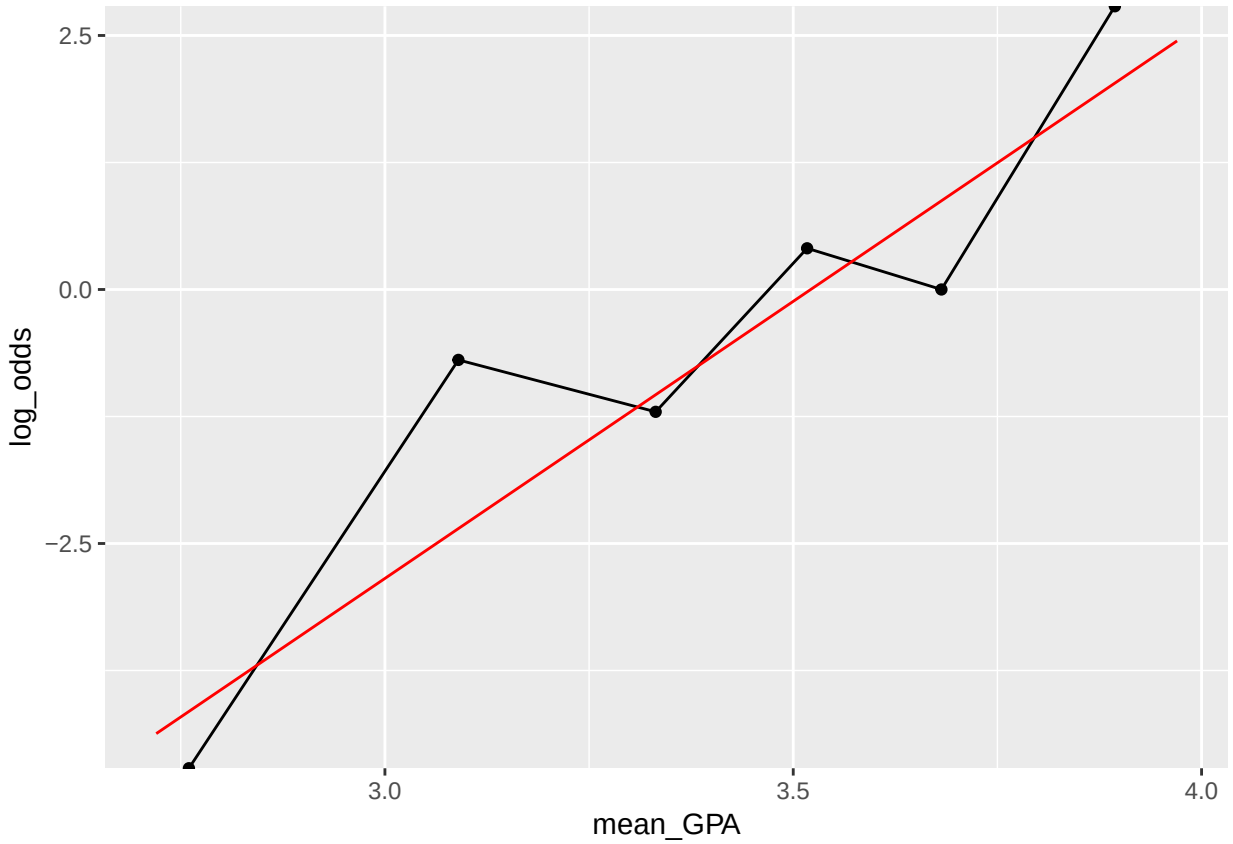



```
#Compute the log of the odds of acceptance
MedGPA_binned<-MedGPA_binned%>%mutate(log_odds=log(acceptance_rate/(1 - acceptance_rate)))

#Plot the log of the odds of acceptance for the binned GPA
data_point<-ggplot(data = MedGPA_binned, aes(x = mean_GPA, y = log_odds)) +geom_point() + geom_line()

# compute log odds for observations
MedGPA_plus<-MedGPA_plus%>%mutate(log_odds_hat = log(.fitted/(1 - .fitted)))

#Plot the logistic model on the log odds scale
data_point+geom_line(data = MedGPA_plus, aes(x = GPA, y = log_odds_hat), color = "red")
```




```
#Create a new data frame
new_data <- data.frame(GPA = 3.51)

#Make predictions for new data frame
augment(mod, newdata = new_data, type.predict = "response")%>%mutate(Acceptance=round(.fitted))
```

```
## # A tibble: 1 x 4
##     GPA .fitted .se.fit Acceptance
##   <dbl>   <dbl>   <dbl>      <dbl>
## 1  3.51   0.484  0.0834          0
```

```
#Convert the data frame with binary predictions
tidy_mod<-augment(mod, type.predict = "response")%>%mutate(Acceptance_hat = round(.fitted))

#Confusion Matrix
tidy_mod%>%select(Acceptance, Acceptance_hat)%>%table()
```

```
##            Acceptance_hat
## Acceptance  0  1
##          0 16  9
##          1  6 24
```