w271: Homework 2 (Due: Week 3) with Suggested Solutions

Professor Jeffrey Yau

Due: Before the Live Session of Week 3

Instructions (Please Read it Carefully!):

- **Page limit of the pdf report: None, but please be reasonable**
- Page setup:
  - Use the following font size, margin, and linespace:
    - fontsize=11pt
    - margin=1in
    - line_spacing=single
- Submission:
  - Homework needs to be completed individually; this is not a group project.
  - Each student submits his/her homework to the course github repo by the deadline; submission and revision made after the deadline will not be graded
  - Submit 2 files:
    1. A pdf file that details your answers. Include all the R codes used to produce the answers. *Please do not suppress the codes in your pdf file.*
    2. R markdown file used to produce the pdf file
  - Use the following file-naming convensation; fail to do so will receive 10% reduction in the grade:
    - StudentFirstNameLastName_HWNumber.fileExtension
    - For example, if the student's name is Kyle Cartman for homework 1, name your files as
      - KyleCartman_HW1.Rmd
      - KyleCartman_HW1.pdf
  - Although it sounds obvious, please write your name on page 1 of your pdf and Rmd files.
  - For statistical methods that we cover in this course, use only the R libraries and functions that are covered in this course. If you use libraries and functions for statistical modeling that we have not covered, you have to (1) provide an explanation of why such libraries and functions are used instead and (2) reference to the library documentation. **Lacking the explanation and reference to the documentation will result in a score of zero for the corresponding question.** For data wrangling and data visualization, you are free to use other libraries, such as dplyr, ggplot2, etc.
  - For mathematical formulae, type them in your R markdown file. **Do not write them on a piece of paper, snap a photo, and either insert the image file or sumbit the image file separately. Doing so will receive a** 0 **for that whole question.**
  - Students are expected to act with regards to UC Berkeley Academic Integrity.

In the live session of week 2, we discussed data analysis, EDA, and binary logistic regression. This homework is designed to review and practice these concepts and techniques. It also covers variable transformation and associated concepts covered in week 3.

For this homework, you will use the dataset *"data_wk02.csv"*, which contains a small sample of graduate school admission data from a university. The variables are specificed below:

1. admit - the depenent variable that takes two values: $0, 1$ where 1 denotes *admitted* and 0 denotes *not admitted*.
2. gre - GRE score
3. gpa - College GPA
4. rank - rank in college major

As some students had questions about "rank" in college major, I want to explain it more. THe variable `rank` represents the "rank", in terms of category, in within a major. Note that these "ranks" are not purely based on GPA within the major; they are also based on students' extra-curricular activities. They are not a prefect preditor of graduate school admission.

Suppose you are hired by the University's Admission Committee and are charged to analyze this data to quantify the effect of GRE, GPA, and college rank on admission probability. We will conduct this analysis by answering the follwing questions:

Question 1: Examine the data and conduct EDA.

```
rm(list = ls())
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)

library(car)
```

```
## Loading required package: carData
```

```
require(dplyr)
```

```
## Loading required package: dplyr
```

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

```
## The following object is masked from 'package:car':
##
##     recode
```

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

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

```r
library(Hmisc)
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
## Loading required package: ggplot2
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
path <- "~/Documents/Teach/Cal/w271/course-main-dev/hw/hw02/soln/"
#path <- "~/Documents/Teach/Cal/w271/_2018.03_Fall/hw/hw02/"
setwd(path)

df <- read.csv("~/Documents/Teach/Cal/w271/course-main-dev/hw/hw02/data_wk02.csv", stringsAsFa
str(df)
```

```
## 'data.frame':    400 obs. of  4 variables:
##  $ admit: int  0 1 1 1 0 1 1 0 1 0 ...
##  $ gre  : int  380 660 800 640 520 760 560 400 540 700 ...
##  $ gpa  : num  3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...
##  $ rank : int  3 3 1 4 4 2 1 2 3 2 ...
```

```r
describe(df)
```

```
## df
##
##  4  Variables      400  Observations
## ---------------------------------------------------------------------------
## admit
##        n  missing distinct     Info      Sum     Mean      Gmd
##      400        0        2     0.65      127   0.3175   0.4345
##
## ---------------------------------------------------------------------------
## gre
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      400        0       26    0.997    587.7    131.2      399      440
##      .25      .50      .75      .90      .95
##      520      580      660      740      800
##
## lowest : 220 300 340 360 380, highest: 720 740 760 780 800
```

```
## ---------------------------------------------------------------------------
## gpa
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      400        0      132        1     3.39   0.4351    2.758    2.900
##      .25      .50      .75      .90      .95
##    3.130    3.395    3.670    3.940    4.000
##
## lowest : 2.26 2.42 2.48 2.52 2.55, highest: 3.95 3.97 3.98 3.99 4.00
## ---------------------------------------------------------------------------
## rank
##        n  missing distinct     Info     Mean      Gmd
##      400        0        4     0.91    2.485    1.038
##
## Value          1     2     3     4
## Frequency     61   151   121    67
## Proportion 0.152 0.378 0.302 0.168
## ---------------------------------------------------------------------------
```

```
table(df$admit)
```

```
##
##   0   1
## 273 127
```

The data set, imported into R as a data.farme called *df*, contains 400 observations and 4 variables. - None of the variables has missing values - Both GRE and GPA are a numeric variables - rank is an ordinal variable - *admit*, which is a binary variable taking values of 0 and 1, is our dependent (or target) variable - all the other three variables, *GRE, GPA, rank*, are potential explanatory variables

Univariate Exploratory Data Analysis

```
# crosstab(df$admit, row.vars = '0/1', col.vars = 'Admit',
# type = 'f')

# Dependent variable: admit
table(df$admit)
```

```
##
##   0   1
## 273 127
```

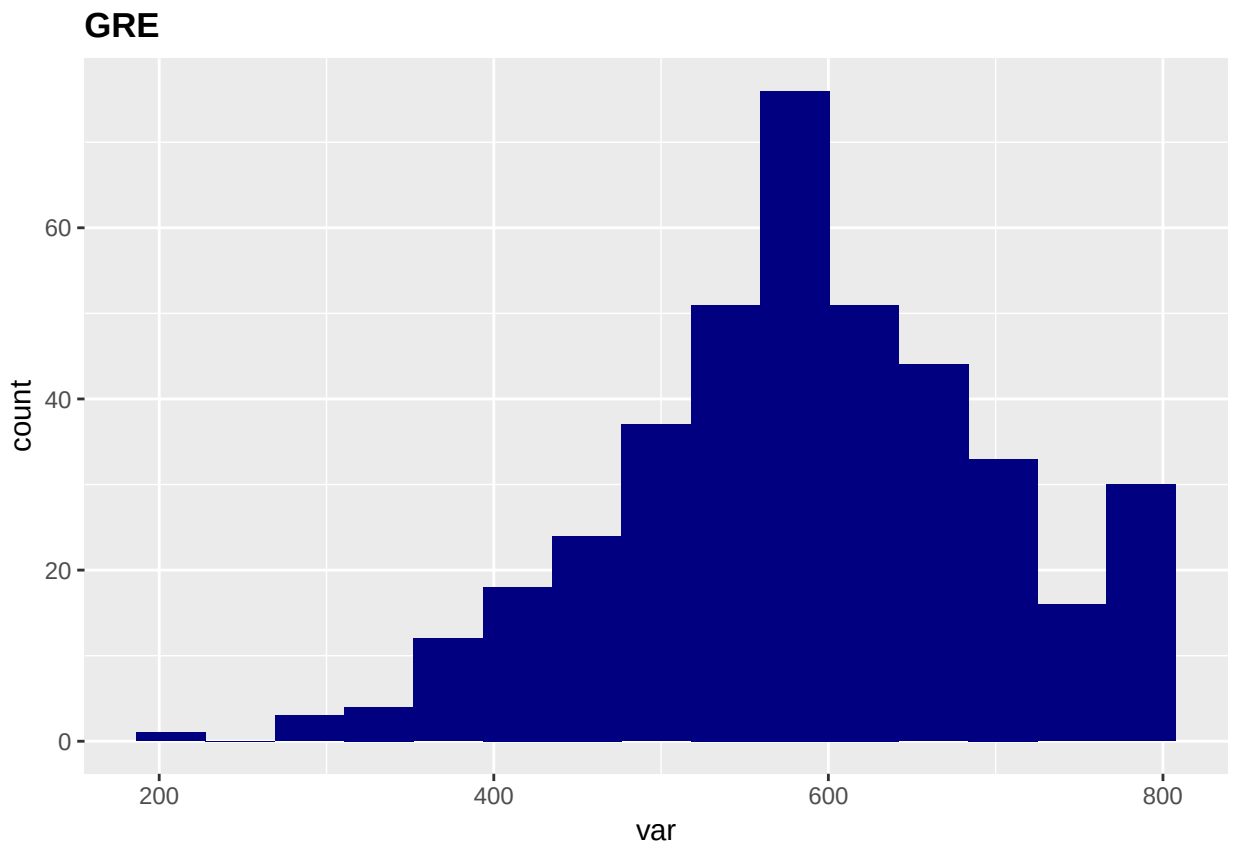
```
prop.table(table(df$admit))
```

```
##
##      0      1
## 0.6825 0.3175
```

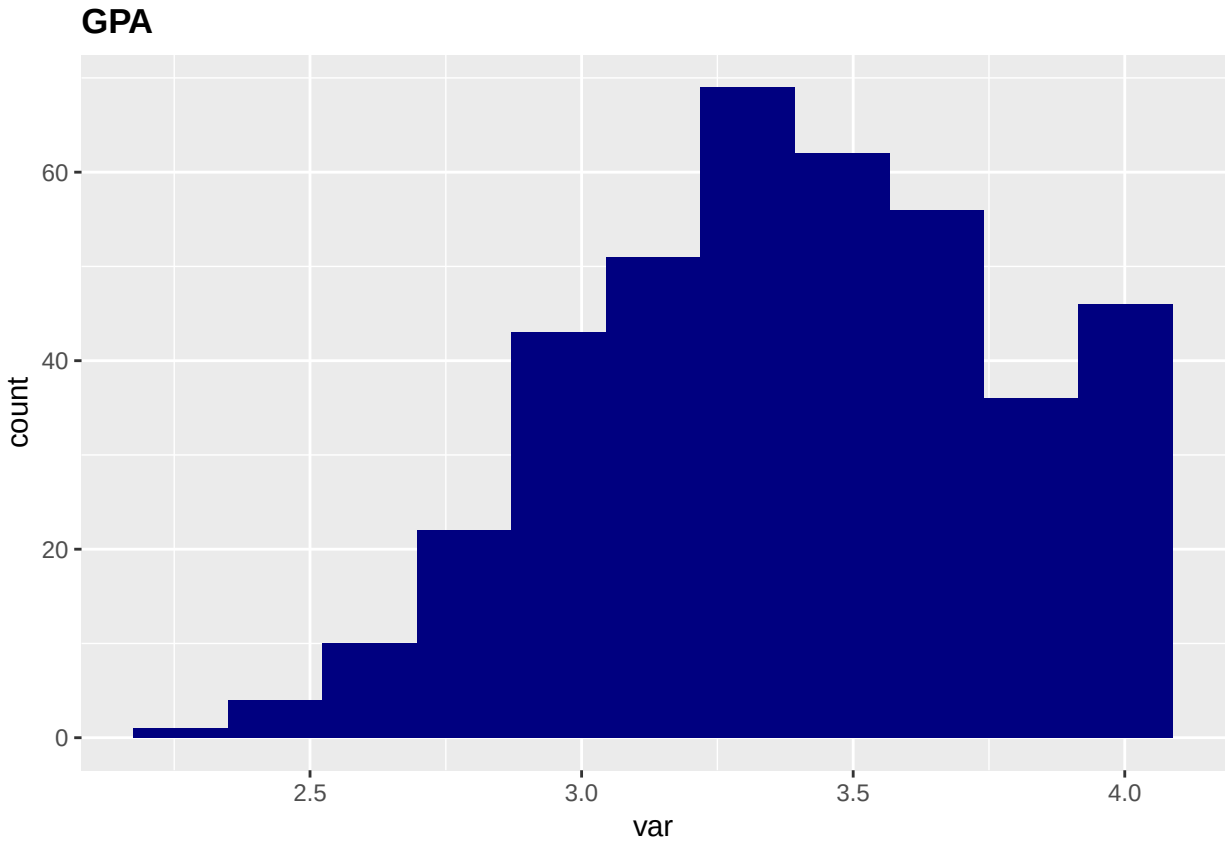
```
# Explanatory Variables:
plot_hist = function(data, var, title) {
```

```
    bw = diff(range(var))/(2 * IQR(var)/length(var)^(1/3))
    p <- ggplot(data, aes(var))
    p + geom_histogram(fill = "navy", bins = bw) + ggtitle(title) +
        theme(plot.title = element_text(lineheight = 1, face = "bold"))
}

# Explanatory Variable: GRE
plot_hist(data = df, var = df$gre, title = "GRE")
```




```
# Explanatory Variable: GPA
plot_hist(data = df, var = df$gpa, title = "GPA")
```




```
# Explanatory Variable: rank
table(df$rank)
```

```
##
##   1   2   3   4
##  61 151 121  67
```

```
round(prop.table(table(df$rank)), 2)
```

```
##
##    1    2    3    4
## 0.15 0.38 0.30 0.17
```

Dependent Variable: admit

The dependent variable, *admit*, is a binary variable taking only values from 0 or 1. Out of 400 students, 237 (or 68.25%) are not admitted and 127 (or 31.75%) are admitted.

Explanatory Variables: GRE and GPA

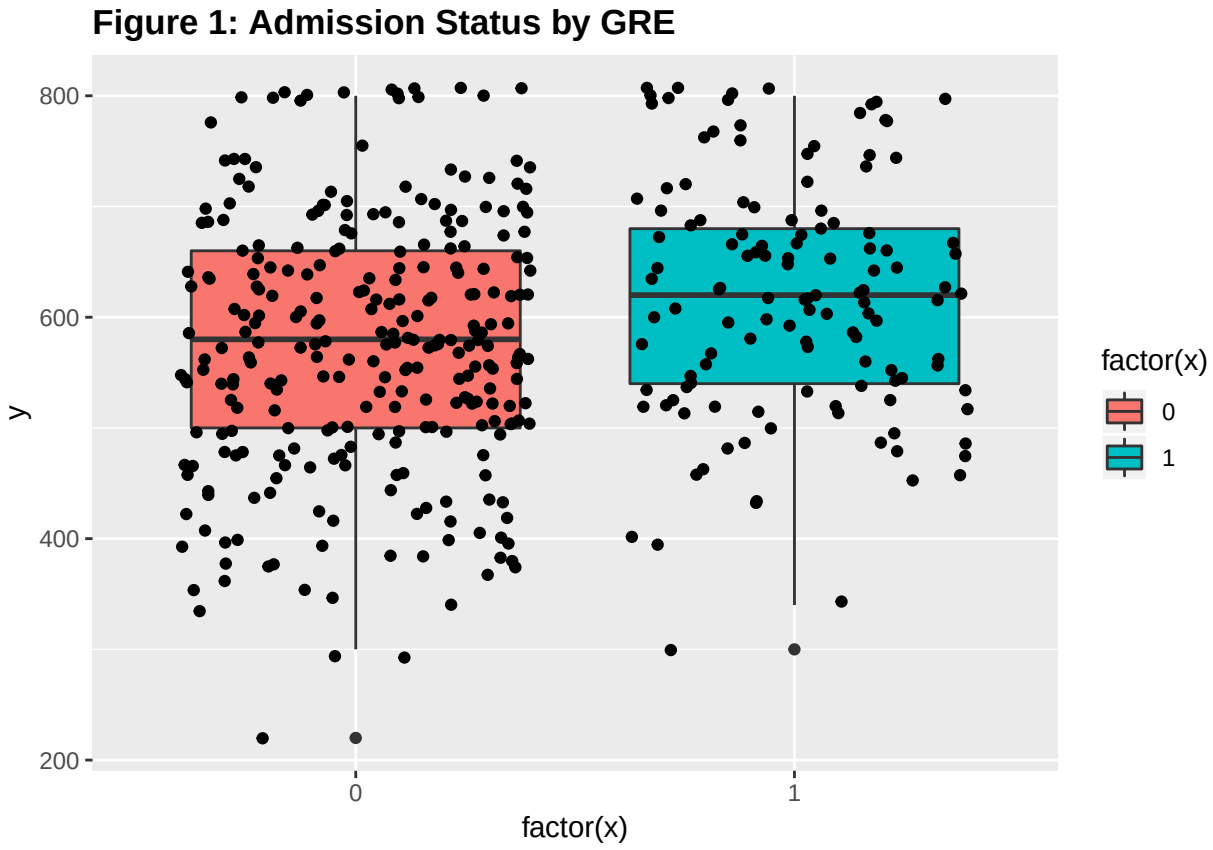
The variable, *GRE*, is a numeric variable that is slightly left-skewed with a mass of observations at 800. For this exercise, I will not transform this variable or bin out the observations at 800. I discussed some of the binning strategies in class.

The variable, *GPA*, is a numeric variable that is left-skewed, with most of the values falling above the value 3.0 and a mass of observations at 4.0. At this point of the analysis, I will not decide whether or not transformation will be conducted.

Bivariate Exploratory Data Analysis

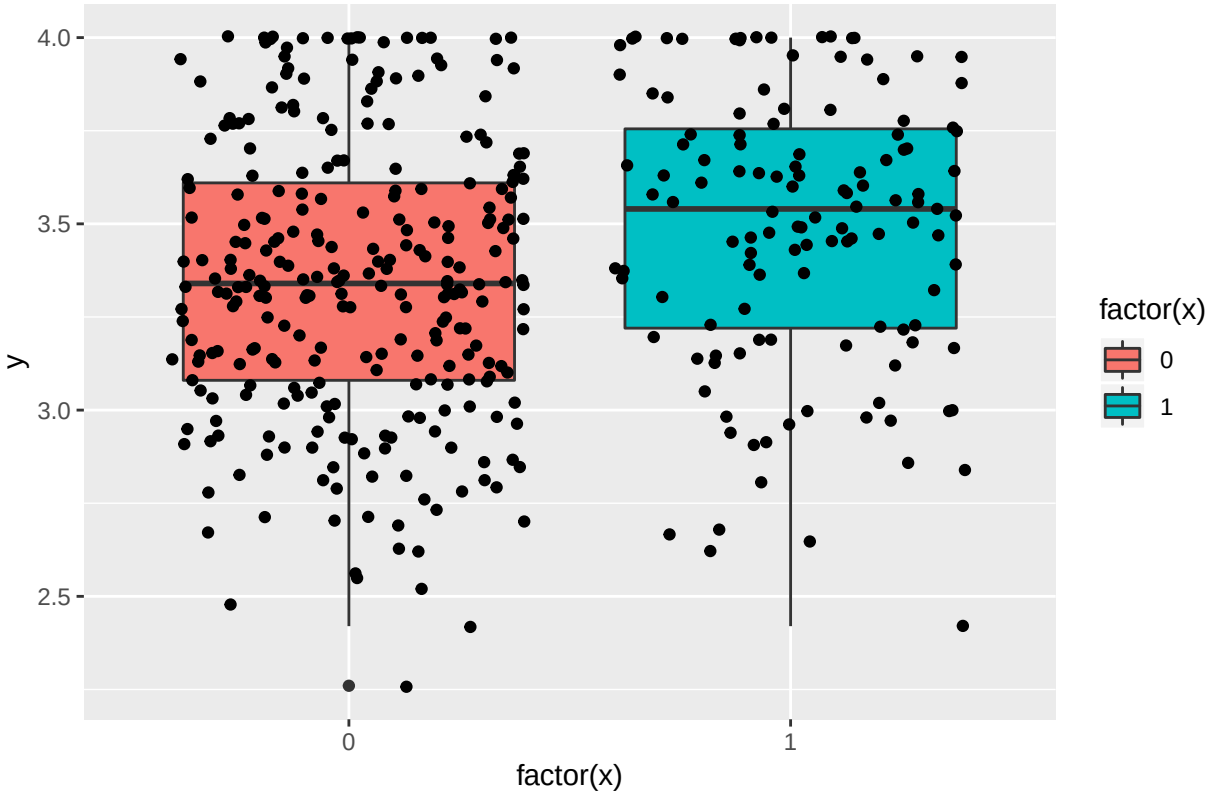
```
plot_box = function(data, x, y, title) {
    ggplot(data, aes(factor(x), y)) + geom_boxplot(aes(fill = factor(x))) +
        geom_jitter() + ggtitle(title) + theme(plot.title = element_text(lineheight = 1,
        face = "bold"))
}

# Admit and GRE
plot_box(df, x = df$admit, y = df$gre, title = "Figure 1: Admission Status by GRE")
```




```
# Admit and GPA
plot_box(df, x = df$admit, y = df$gpa, title = "Figure 2: Admission Status by GPA")
```

Figure 2: Admission Status by GPA




```
# Admit and Rank
xtabs(~df$admit + df$rank)
```

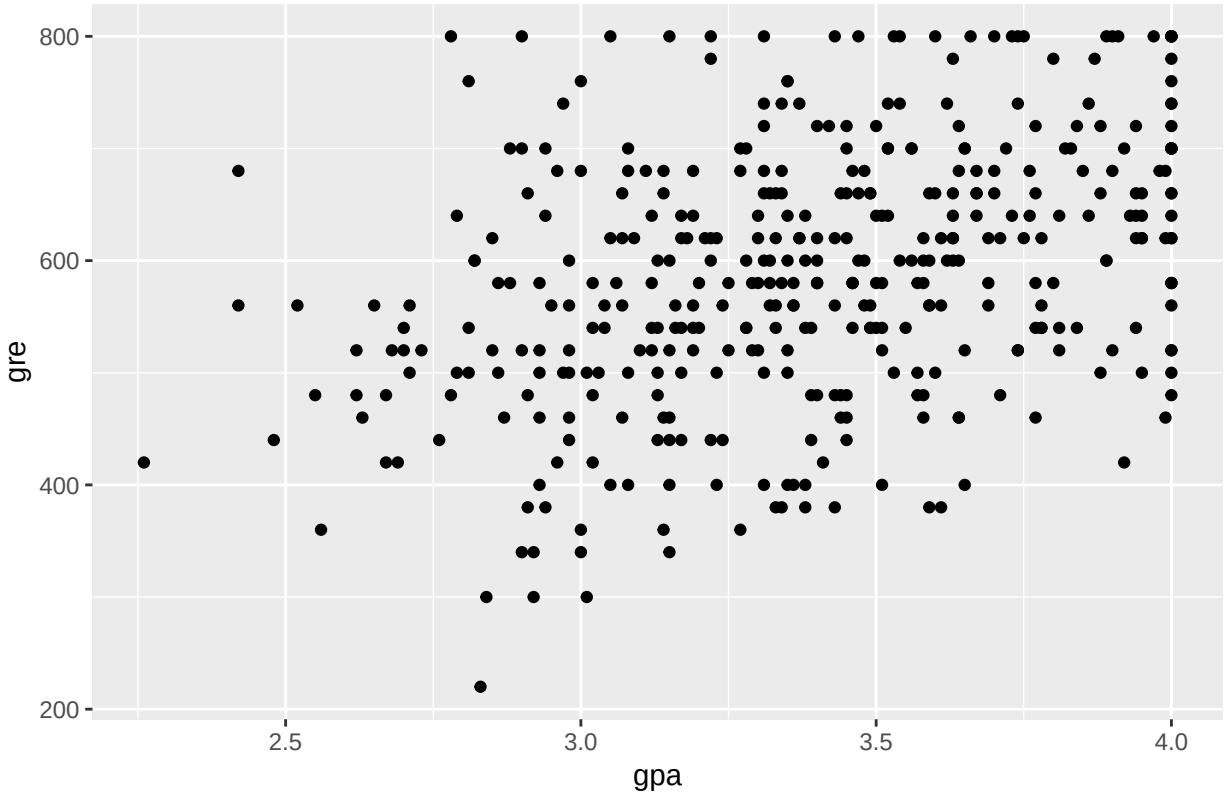
```
##          df$rank
## df$admit  1  2  3  4
##        0 28 97 93 55
##        1 33 54 28 12
```

```
round(prop.table(xtabs(~df$admit + df$rank), 2), 2)
```

```
##          df$rank
## df$admit    1    2    3    4
##        0 0.46 0.64 0.77 0.82
##        1 0.54 0.36 0.23 0.18
```

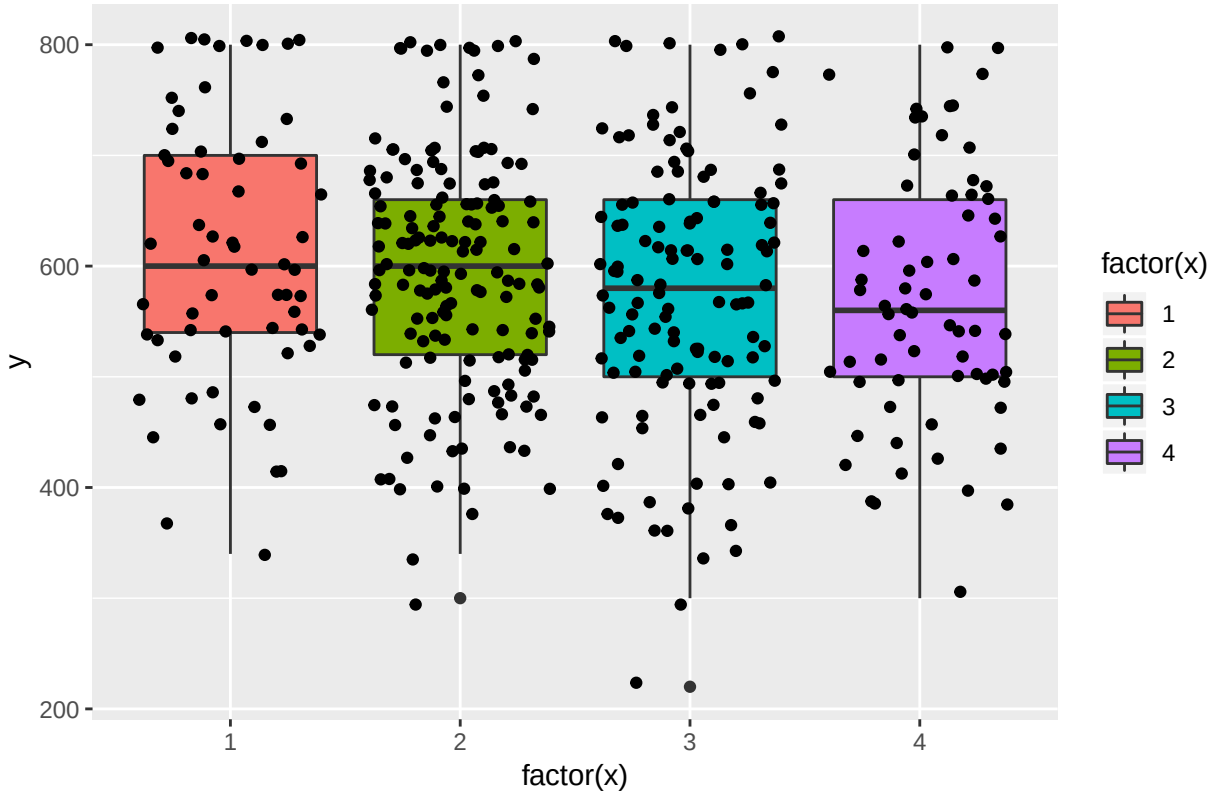
```
# GRE and GPA
p <- ggplot(df, aes(gpa, gre))
p + geom_point() + ggtitle("Figure 3: GRE vs GPA") + theme(plot.title = element_text(lineheigh
    face = "bold"))
```

Figure 3: GRE vs GPA



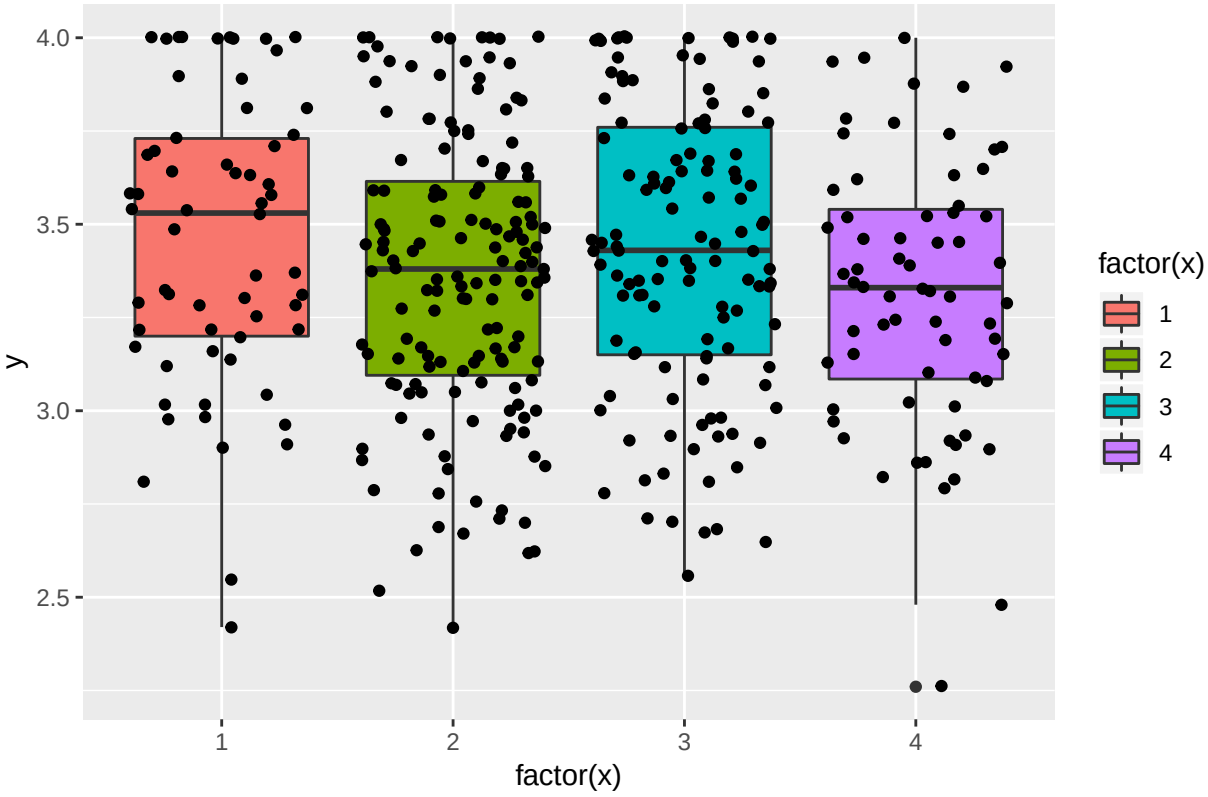

```
# GRE and Rank
plot_box(df, x = df$rank, y = df$gre, title = "Figure 4: GRE by Rank")
```

Figure 4: GRE by Rank




```
# GPA and Rank
plot_box(df, x = df$rank, y = df$gpa, title = "Figure 5: GPA by Rank")
```

Figure 5: GPA by Rank



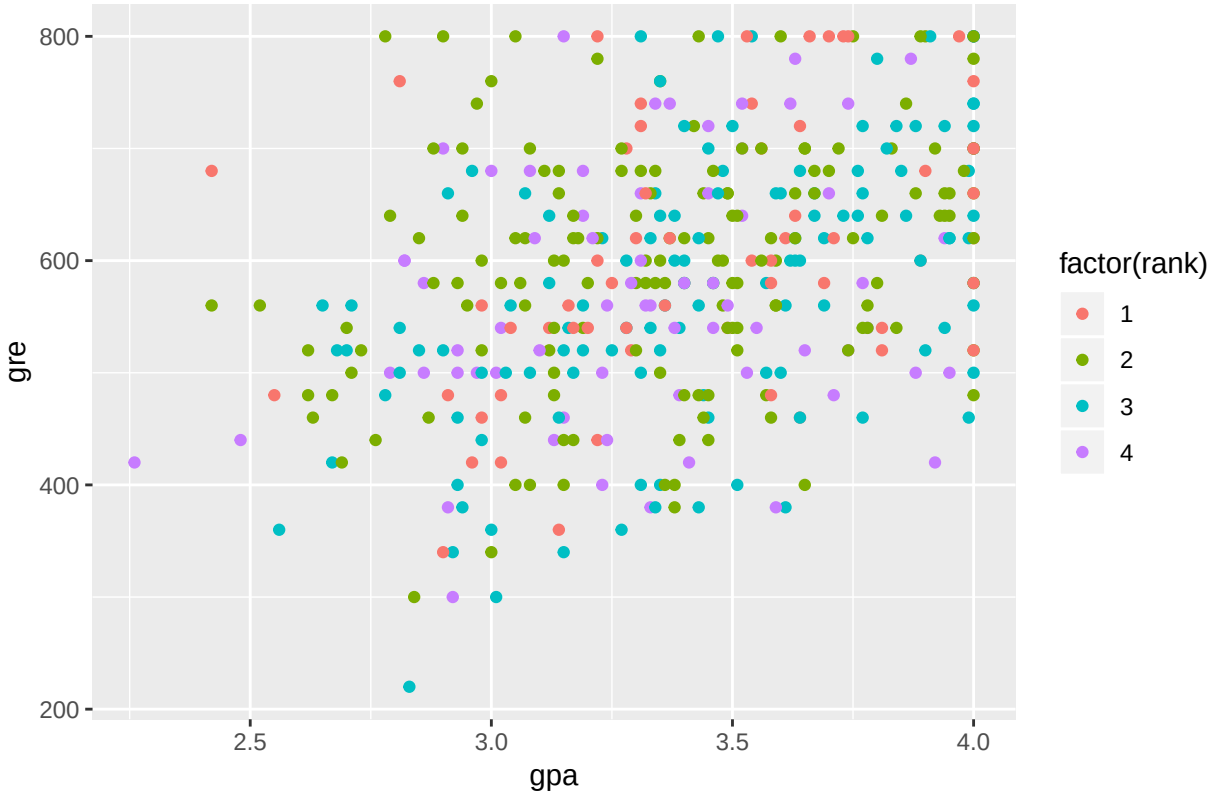
From the bivariate analysis, students who were admitted, not surprisingly, tend to have higher GPA and GPA (Figure 1 and 2), and students who had higher GPA also tended to have higher GRE scorer, as shown in Figure 3. I said "tend to" because there were admitted students who had low GPA. In fact, taking pretty much any value of GPA, there were students who were admitted and students who did not.

There also a strong bivariate relationship between rank and admit: as the rank went down, admission rate also went down, as shown in the two frequency tables. However, while students with higher rank (i.e. lower rank value - e.g. rank 1 is "higher" than rank 2) had higher a GRE score (Figure 4), but there was no clear relationship between rank and GPA (Figure 5), disputing the fact that rank is a monotonic function of GPA.

Multivariate Exploratory Data Analysis

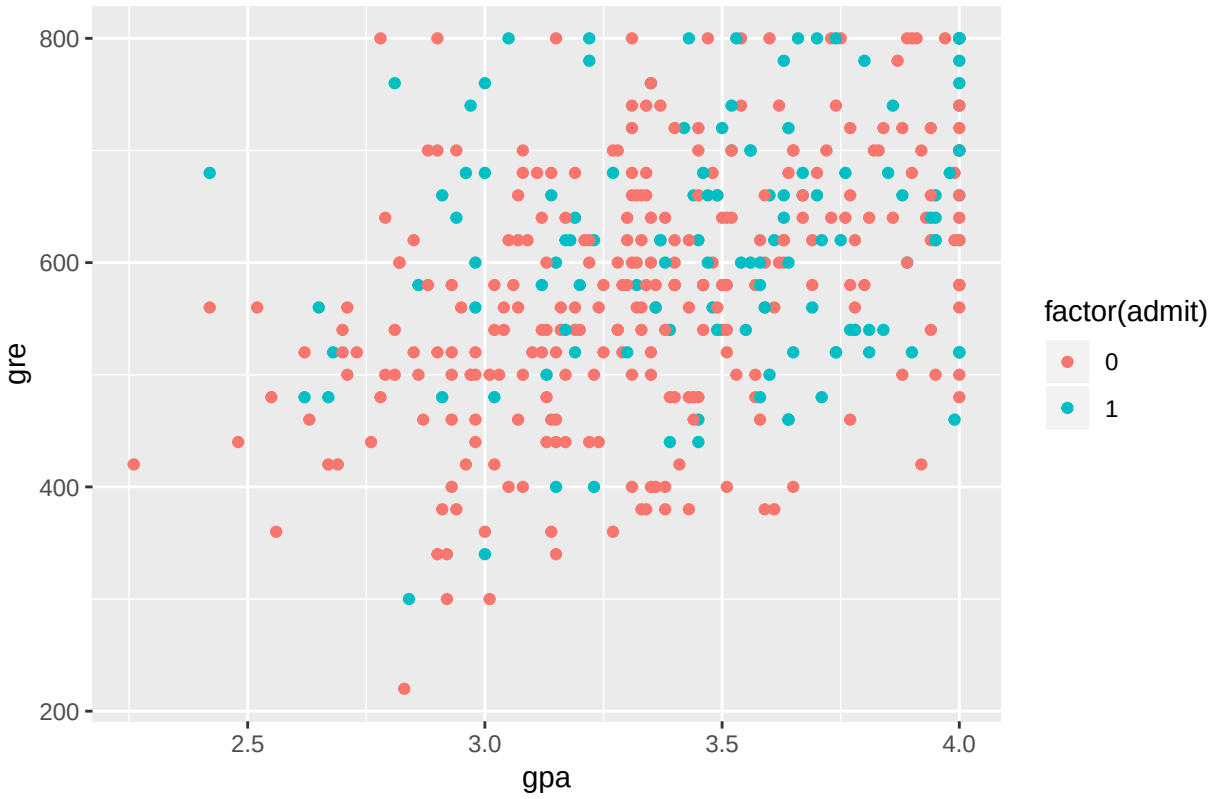
```
# GRE, GPA, and Rank
p <- ggplot(df, aes(gpa, gre))
p + geom_point(aes(colour = factor(rank))) + ggtitle("Figure 6: GRE vs GPA colored by Rank") +
    theme(plot.title = element_text(lineheight = 1, face = "bold"))
```

Figure 6: GRE vs GPA colored by Rank




```
# Admit, GRE, and GPA
p <- ggplot(df, aes(gpa, gre))
p + geom_point(aes(colour = factor(admit))) + ggtitle("Figure 7: GRE vs GPA colored by Admit")
    theme(plot.title = element_text(lineheight = 1, face = "bold"))
```

Figure 7: GRE vs GPA colored by Admit



From Figure 6, it is not easy to detect whether students with high GPA and GRE also were highly ranked, though students with low GPA and low GRE tended to be in rank 3 and 4.

It is also hard to definitely conclude the position relationship between admission and high GRE and GPA (Figure 7).

Question 2: Estimate a binary logistic regression using the following set of explanatory variables: gre, gpa, $rank$, gre^2, gpa^2, and $gre \times gpa$, where $gre \times gpa$ denotes the interaction between gre and gpa variables.

```
admit.glm1 <- glm(admit ~ gre + gpa + rank + I(gre^2) + I(gpa^2) +
    gre:gpa, family = binomial, data = df)
summary(admit.glm1)
```

```
##
## Call:
## glm(formula = admit ~ gre + gpa + rank + I(gre^2) + I(gpa^2) +
##     gre:gpa, family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4928  -0.8958  -0.6192   1.1436   2.2211
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -7.092e+00  9.024e+00  -0.786   0.4319
## gre          1.845e-02  1.169e-02   1.578   0.1146
## gpa         -7.960e-03  4.933e+00  -0.002   0.9987
## rank        -5.643e-01  1.278e-01  -4.414 1.01e-05 ***
## I(gre^2)     3.495e-06  8.156e-06   0.429   0.6683
## I(gpa^2)     6.511e-01  7.605e-01   0.856   0.3919
## gre:gpa     -5.987e-03  3.186e-03  -1.879   0.0602 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 455.72  on 393  degrees of freedom
## AIC: 469.72
##
## Number of Fisher Scoring iterations: 4
```

```
round(exp(cbind(Estimate = coef(admit.glm1), confint(admit.glm1))),
    2)
```

```
## Waiting for profiling to be done...
```

```
##             Estimate 2.5 %   97.5 %
## (Intercept)     0.00  0.00 20611.25
## gre             1.02  1.00     1.04
## gpa             0.99  0.00 21913.15
## rank            0.57  0.44     0.73
## I(gre^2)        1.00  1.00     1.00
## I(gpa^2)        1.92  0.42     8.44
## gre:gpa         0.99  0.99     1.00
```

```
vcov(admit.glm1)
```

```
##               (Intercept)           gre           gpa          rank
## (Intercept)  8.142830e+01 -4.029898e-02 -4.092968e+01  8.991671e-03
## gre         -4.029898e-02  1.366884e-04 -2.647056e-04 -2.184187e-05
## gpa         -4.092968e+01 -2.647056e-04  2.433520e+01 -2.286230e-02
## rank         8.991671e-03 -2.184187e-05 -2.286230e-02  1.634106e-02
## I(gre^2)     6.995806e-06 -4.711986e-08  3.578829e-06 -2.749812e-08
## I(gpa^2)     5.180618e+00  2.082437e-03 -3.471642e+00  1.329532e-03
## gre:gpa      9.124435e-03 -2.319674e-05 -1.134051e-03  1.892281e-05
##                  I(gre^2)      I(gpa^2)       gre:gpa
## (Intercept)  6.995806e-06  5.180618e+00  9.124435e-03
## gre         -4.711986e-08  2.082437e-03 -2.319674e-05
## gpa          3.578829e-06 -3.471642e+00 -1.134051e-03
## rank        -2.749812e-08  1.329532e-03  1.892281e-05
## I(gre^2)     6.651608e-11  3.404387e-07 -9.531835e-09
## I(gpa^2)     3.404387e-07  5.783635e-01 -7.414826e-04
## gre:gpa     -9.531835e-09 -7.414826e-04  1.014993e-05
```

Question 3: Test the hypothesis that GRE has no effect on admission using the likelihood ratio test.

```
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
# Estimate the model under the null hypothesis
admit.glm1.h0 <- glm(admit ~ gpa + rank + I(gpa^2), family = binomial,
    data = df)
# Estiamte the modle under the alternative hypothesis
admit.glm1.h1 <- glm(admit ~ gre + gpa + rank + I(gre^2) + I(gpa^2) +
    gre:gpa, family = binomial, data = df)
# Though not required, it's a good practice to display the
# model results side-by-side
stargazer(admit.glm1.h0, admit.glm1.h1, type = "text")
```

```
##
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                                admit
##                           (1)            (2)
## ------------------------------------------------
## gre                                      0.018
##                                         (0.012)
##
## gpa                      -0.514         -0.008
##                         (4.769)        (4.933)
##
## rank                   -0.581***      -0.564***
##                         (0.126)        (0.128)
##
## I(gre2)                                 0.00000
##                                        (0.00001)
##
## I(gpa2)                  0.228          0.651
##                         (0.703)        (0.761)
##
## gre:gpa                                -0.006*
##                                        (0.003)
##
## Constant                 -0.306         -7.092
##                         (8.024)        (9.024)
##
```

```
## ------------------------------------------------
## Observations                400             400
## Log Likelihood           -231.915        -227.861
## Akaike Inf. Crit.        471.830         469.723
## ================================================
## Note:                  *p<0.1; **p<0.05; ***p<0.01
```

```
# Test the hypothesis
anova(admit.glm1.h0, admit.glm1.h1)
```

```
## Analysis of Deviance Table
##
## Model 1: admit ~ gpa + rank + I(gpa^2)
## Model 2: admit ~ gre + gpa + rank + I(gre^2) + I(gpa^2) + gre:gpa
##   Resid. Df Resid. Dev Df Deviance
## 1       396     463.83
## 2       393     455.72  3   8.1071
```

```
anova(admit.glm1.h0, admit.glm1.h1)$Df
```

```
## [1] NA  3
```

```
# Calculate p-value
pvalue <- 1 - pchisq(q = anova(admit.glm1.h0, admit.glm1.h1)$Deviance,
    df = anova(admit.glm1.h0, admit.glm1.h1)$Df)
pvalue
```

```
## [1]        NA 0.0438492
```

As p-value = 0.044, which is under 0.05, the hypothesis is rejected. GRE has no effect on admission in the presence of GPA.

Question 4: What is the estimated effect of college GPA on admission?

Since we reject the model under the null hypothesis, we will use the model under the alternative hypothesis for the estimated effect of GPA on admission.

The estimated model is

$$logit(\hat{\pi}) = -7.092 + 0.0185GRE - 0.0080GPA - 0.5643rank + 0.0GRE^2 + 0.65GPA^2 - 0.0060GRE * GPA$$

or

$$\hat{\pi} = exp(-7.092 + 0.0185GRE - 0.0080GPA - 0.5643rank + 0.0GRE^2 + 0.65GPA^2 - 0.0060GRE * GPA)$$

The estimated effect on the odds of admission when GPA change by k units of GPA is

$$\begin{aligned}\widehat{OR} &= \frac{Odds_{GPA+k}}{Odds_{GPA}} \\ &= \frac{exp(-7.092 + 0.0185GRE - 0.0080(GPA + k) - 0.5643rank + 0.0GRE^2 + 0.65(GPA + k)^2 - 0.0060GRE}{exp(-7.092 + 0.0185GRE - 0.0080GPA - 0.5643rank + 0.0GRE^2 + 0.65GPA^2 - 0.0060GRE * G} \\ &= exp(-0.0080k + 2 \times 0.65k - 0.0060k * GRE)\end{aligned}$$

Due to the quadratic term associated with GPA and the interaction between GRE and GPA, the estimated effect on admission of GPA is a function of both the GPA and GRE. For instance, for $k = 0.5$, $GPA = 3.0$, and $GRE = 600$, the odds is estimated to increated by 16.6%. Note that the estimated increase in odds of a 0.5% increase in GPA is much larger (i.e. 61.4%) for someone with GPA of 3.5 and GRE of 600.

The calculation is detailed below.

```
impact_GPA = function(k, GRE, GPA) {
    exp(admit.glm1.h1$coefficients["gpa"] * k + 2 * k * admit.glm1.h1$coefficients["I(gpa^2)"]
        GPA + admit.glm1.h1$coefficients["gre:gpa"] * k * GRE)
}

impact_GPA(k = 0.5, GRE = 600, GPA = 3)
```

```
##      gpa
## 1.165583
```

```
impact_GPA(k = 0.5, GRE = 600, GPA = 3.5)
```

```
##      gpa
## 1.614059
```

```
# Calculate a range of GPA effects, holding GRE=600
GPA = seq(from = 2.8, to = 4, by = 0.1)

data.frame(GPA = GPA, GRE = 600, GPA_effect = impact_GPA(k = 0.1,
    GRE = 600, GPA = GPA))
```

```
##    GPA GRE GPA_effect
## 1  2.8 600   1.004613
## 2  2.9 600   1.017779
## 3  3.0 600   1.031119
## 4  3.1 600   1.044633
## 5  3.2 600   1.058324
## 6  3.3 600   1.072195
## 7  3.4 600   1.086248
## 8  3.5 600   1.100484
## 9  3.6 600   1.114908
## 10 3.7 600   1.129520
## 11 3.8 600   1.144324
## 12 3.9 600   1.159322
## 13 4.0 600   1.174516
```

```
# Calculate a range of GPA effects, holding GRE=750
GPA = seq(from = 2.8, to = 4, by = 0.1)

data.frame(GPA = GPA, GRE = 750, GPA_effect = impact_GPA(k = 0.1,
    GRE = 750, GPA = GPA))
```

```
##    GPA GRE GPA_effect
```

```
## 1  2.8 750  0.9183312
## 2  2.9 750  0.9303672
## 3  3.0 750  0.9425609
## 4  3.1 750  0.9549145
## 5  3.2 750  0.9674299
## 6  3.3 750  0.9801094
## 7  3.4 750  0.9929551
## 8  3.5 750  1.0059691
## 9  3.6 750  1.0191537
## 10 3.7 750  1.0325111
## 11 3.8 750  1.0460436
## 12 3.9 750  1.0597534
## 13 4.0 750  1.0736429
```

Question 5: Construct the confidence interval for the admission probability for the students with $GPA = 3.3$, $GRE = 720$, and $rank = 1$.

```
gpa = 3.3
gre = 720
rank = 1
predict.data = data.frame(intercept = 1, gre = gre, gpa = gpa,
    rank = rank, gre_sq = gre^2, gpa_sq = gpa^2, gre_gpa = gre *
        gpa)

predict(object = admit.glm1, newdata = predict.data, type = "link")
```

```
##         1
## 0.2789539
```

```
pi.hat = predict(object = admit.glm1, newdata = predict.data,
    type = "response")
round(pi.hat, 2)
```

```
##    1
## 0.57
```

```
library(mcprofile)
K = matrix(data = c(1, gre, gpa, rank, gre^2, gpa^2, gre * gpa),
    nrow = 1, ncol = length(admit.glm1$coefficients))
# Calculate -2log(Lambda)
linear.combo <- mcprofile(object = admit.glm1, CM = K)
# CI for linear combo
ci.logit.profile <- confint(object = linear.combo, level = 0.95)
# CI for pi.hat
round(exp(ci.logit.profile$confint)/(1 + exp(ci.logit.profile$confint)),
    4)
```

```
##    lower  upper
## 1 0.4373 0.6938
```

For the students with $GPA = 3.3$, $GRE = 720$, and $rank = 1$, the $\hat{\pi} = 0.57$ (or 57%) and the the

Profile LR interval is $[0.4373,0.6938]