Logistic Regression Applications Solutions

Lt Col Ken Horton Lt Col Kris Pruitt Professor Bradley Warner

11 December, 2020

Exercises

1. Possum classification

Let's investigate the `possum` data set again. This time we want to model a binary outcome variable. As a reminder, the common brushtail possum of the Australia region is a bit cuter than its distant cousin, the American opossum. We consider 104 brushtail possums from two regions in Australia, where the possums may be considered a random sample from the population. The first region is Victoria, which is in the eastern half of Australia and traverses the southern coast. The second region consists of New South Wales and Queensland, which make up eastern and northeastern Australia.

We use logistic regression to differentiate between possums in these two regions. The outcome variable, called `pop`, takes value `Vic` when a possum is from Victoria and `other` when it is from New South Wales or Queensland. We consider five predictors: `sex`, `head_l`, `skull_w`, `total_l`, and `tail_l`.

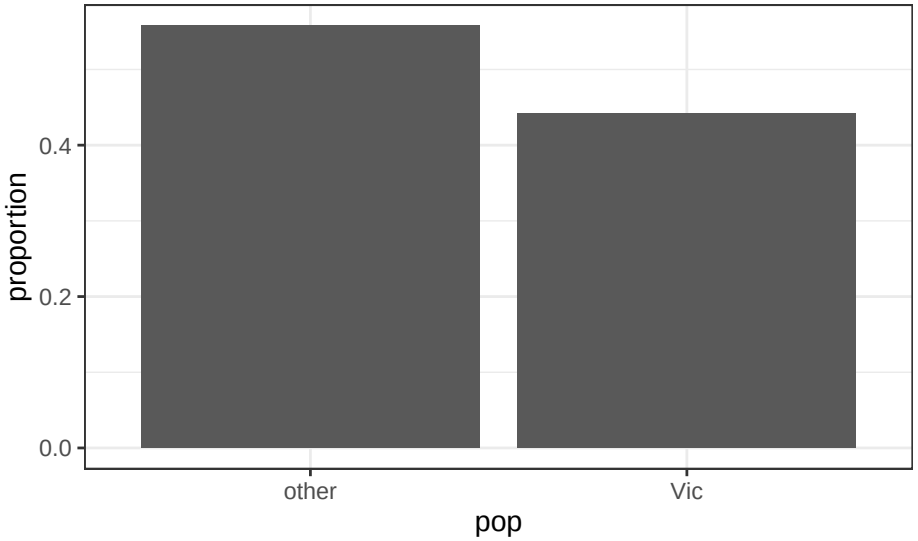
a. Explore the data by making histograms of the quantitative variables, and bar charts of the discrete variables. Are there any outliers that are likely to have a very large influence on the logistic regression model?

```
possum <- read_csv("data/possum.csv") %>%
  select(pop,sex,head_l,skull_w,total_l,tail_l) %>%
  mutate(pop=factor(pop),sex=factor(sex))
```

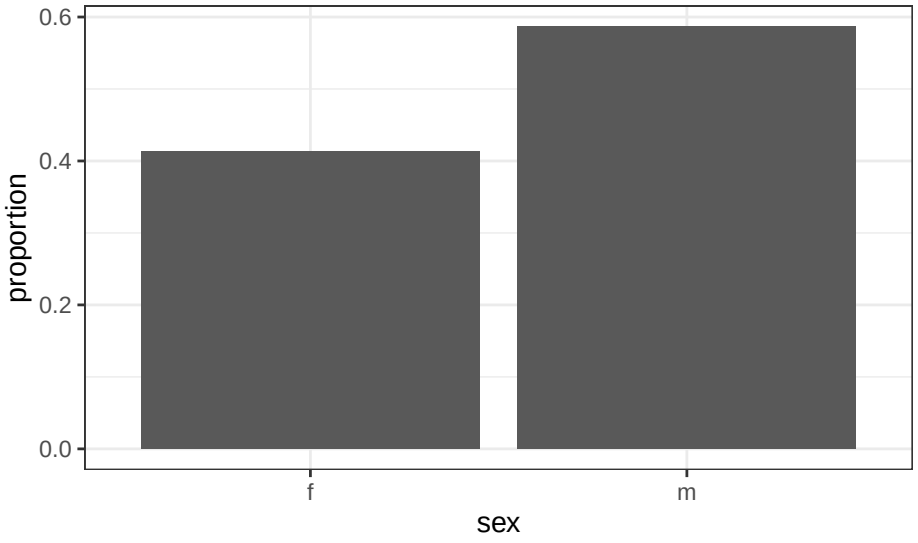
```
inspect(possum)
```

```
##
## categorical variables:
##   name  class levels   n missing                                   distribution
## 1  pop factor      2 104       0 other (55.8%), Vic (44.2%)
## 2  sex factor      2 104       0 m (58.7%), f (41.3%)
##
## quantitative variables:
##         name   class  min     Q1 median     Q3   max     mean       sd   n
## ...1  head_l numeric 82.5 90.675  92.80 94.725 103.1 92.60288 3.573349 104
## ...2 skull_w numeric 50.0 54.975  56.35 58.100  68.6 56.88365 3.113426 104
## ...3 total_l numeric 75.0 84.000  88.00 90.000  96.5 87.08846 4.310549 104
## ...4  tail_l numeric 32.0 35.875  37.00 38.000  43.0 37.00962 1.959518 104
##      missing
## ...1       0
## ...2       0
## ...3       0
## ...4       0
```

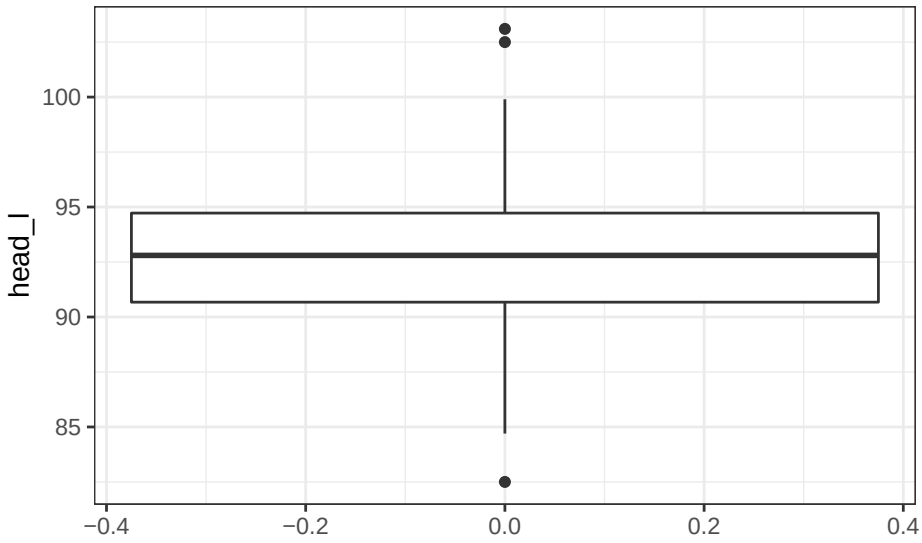
```
possum %>%
  gf_props(~pop) %>%
  gf_theme(theme_bw())
```



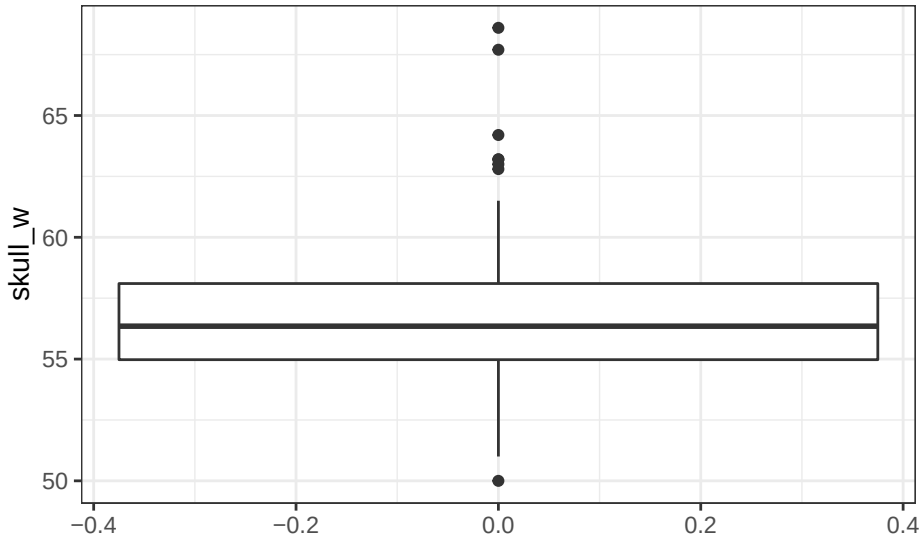

```
possum %>%
  gf_props(~sex) %>%
  gf_theme(theme_bw())
```



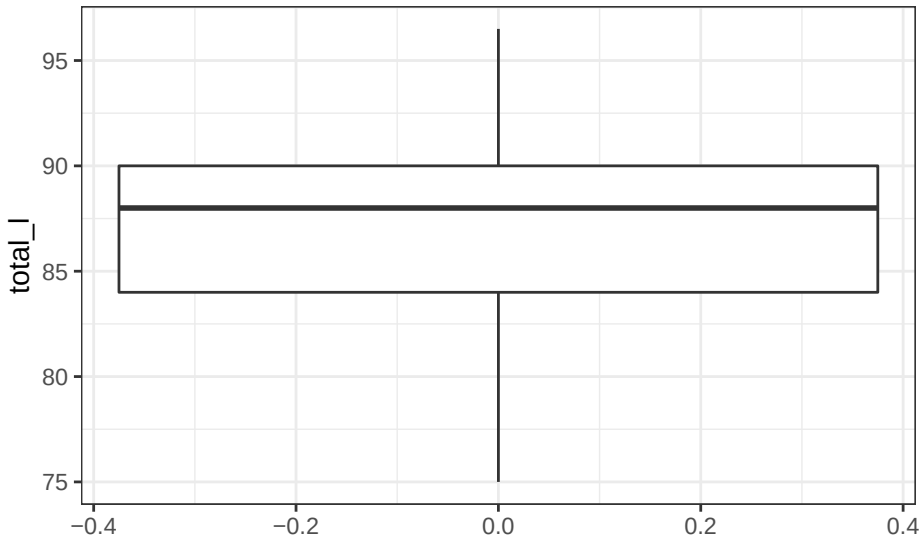

```
possum %>%
  gf_boxplot(~head_l) %>%
  gf_theme(theme_bw())
```



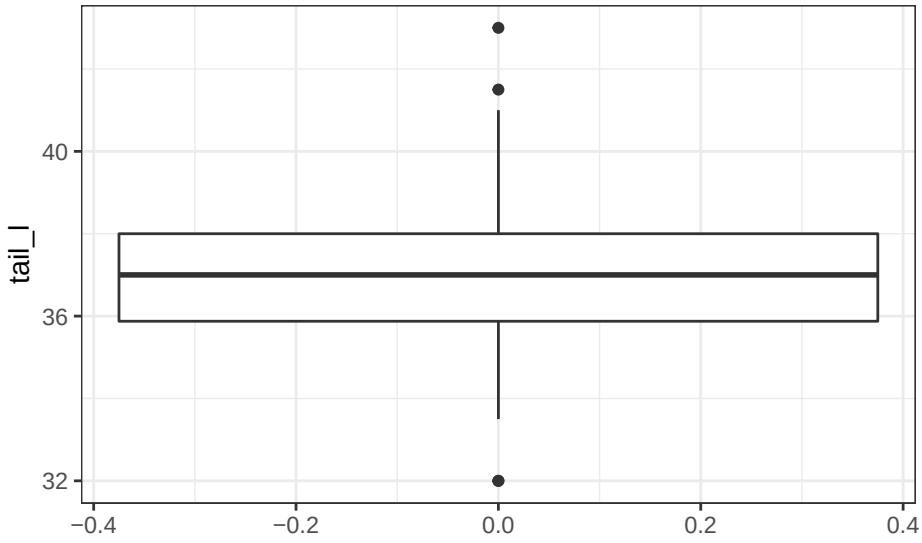

```
possum %>%
  gf_boxplot(~skull_w) %>%
  gf_theme(theme_bw())
```




```
possum %>%
  gf_boxplot(~total_l) %>%
  gf_theme(theme_bw())
```

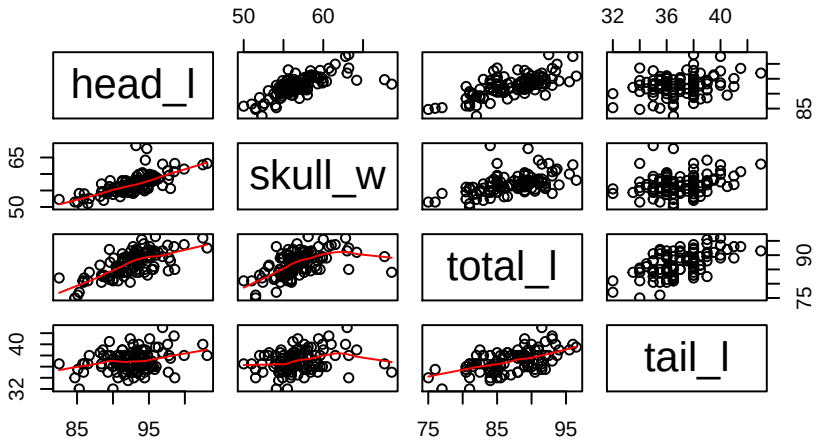



```
possum %>%
  gf_boxplot(~tail_l) %>%
  gf_theme(theme_bw())
```



There are some potential outliers for skull width but otherwise not much concern.

```
pairs(possum[,3:6],lower.panel = panel.smooth)
```



We can see that `head_l` is correlated with the other three variables. This will cause some multicollinearity problems.

b. Build a logistic regression model with all the variable. Report a summary of the model.

```
possum_mod <- glm(pop=="Vic"~.,data=possum,family="binomial")
```

```
summary(possum_mod)
```

```
##
## Call:
## glm(formula = pop == "Vic" ~ ., family = "binomial", data = possum)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6430  -0.5514  -0.1182   0.3760   2.8501
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  39.2349    11.5368   3.401 0.000672 ***
## sexm         -1.2376     0.6662  -1.858 0.063195 .
## head_l       -0.1601     0.1386  -1.155 0.248002
## skull_w      -0.2012     0.1327  -1.517 0.129380
## total_l       0.6488     0.1531   4.236 2.27e-05 ***
## tail_l       -1.8708     0.3741  -5.001 5.71e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 142.787  on 103  degrees of freedom
## Residual deviance:  72.155  on  98  degrees of freedom
## AIC: 84.155
##
## Number of Fisher Scoring iterations: 6
```

```
confint(possum_mod)
```

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

```
##                  2.5 %      97.5 %
## (Intercept) 18.8530781 64.66444839
## sexm        -2.6227018  0.02472167
## head_l      -0.4428559  0.10865739
## skull_w     -0.4933140  0.04479826
## total_l      0.3768179  0.98455786
## tail_l      -2.7170468 -1.23231969
```

c. Using the p-values decide if you want to remove a variable(S) and if so build that model.

Let's remove `head_l` first.

```
possum_mod_red <- glm(pop=="Vic"~sex+skull_w+total_l+tail_l,data=possum,family="binomial")
```

```
summary(possum_mod_red)
```

```
##
## Call:
## glm(formula = pop == "Vic" ~ sex + skull_w + total_l + tail_l,
##     family = "binomial", data = possum)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8102  -0.5683  -0.1222   0.4153   2.7599
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  33.5095     9.9053   3.383 0.000717 ***
## sexm         -1.4207     0.6457  -2.200 0.027790 *
## skull_w      -0.2787     0.1226  -2.273 0.023053 *
## total_l       0.5687     0.1322   4.302 1.69e-05 ***
## tail_l       -1.8057     0.3599  -5.016 5.26e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 142.787  on 103  degrees of freedom
## Residual deviance:  73.516  on  99  degrees of freedom
## AIC: 83.516
##
## Number of Fisher Scoring iterations: 6
```

Since `head_l` was correlated with the other variables, removing it has increased the precision, decreased the standard error, of the other predictors. There p-values are all now less than 0.05.

d. For any variable you decide to remove, build a 95% confidence interval for the parameter.

```
confint(possum_mod)
```

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

```
##                  2.5 %      97.5 %
## (Intercept) 18.8530781 64.66444839
## sexm        -2.6227018  0.02472167
## head_l      -0.4428559  0.10865739
## skull_w     -0.4933140  0.04479826
## total_l      0.3768179  0.98455786
## tail_l      -2.7170468 -1.23231969
```

We are 95% confident that the true slope coefficient for `head_l` is between -0.44 and 0.108.

The bootstrap is not working for this problem. It may be that we have convergence issues when we resample the data. This is a reminder that we need to be careful and not just run methods without checking results. Here is the code:

```
set.seed(952)
results<-do(1000)*glm(pop=="Vic"~.,data=resample(possum),family="binomial")
```

```
head(results[,1:5])
```

```
##     Intercept          sexm        head_l       skull_w       total_l
## 1 -1184.61875  2.122389e+01  3.861561e+00  2.749263e+00  7.274005e+00
## 2  6371.55550 -1.301514e+02  1.023732e+01 -2.738816e+01 -1.076913e+01
## 3 -9612.61941 -2.392900e+03 -1.875252e+02  5.782027e+02 -2.820691e+02
## 4   -25.18662 -1.852185e+01  2.097593e+01 -1.353619e+01  1.483815e+01
## 5   -26.56607 -1.398995e-14 -2.258909e-14  6.528545e-15  2.388756e-14
## 6 -1025.00035  6.159665e+01  2.526181e+01 -2.032143e+01  1.438639e+01
```

```
confint(results)
```

```
##        name       lower      upper level     method   estimate
## 1 Intercept -8030.43219 8566.11832  0.95 percentile 39.2349178
## 2      sexm  -201.28404  207.55196  0.95 percentile -1.2375895
## 3    head_l  -122.70294   63.61867  0.95 percentile -0.1600622
## 4   skull_w   -35.94883   92.78429  0.95 percentile -0.2012445
## 5   total_l   -32.84729   87.94284  0.95 percentile  0.6488131
## 6    tail_l  -138.14608  151.97362  0.95 percentile -1.8708001
```

These intervals are much too large.

e. Explain why the remaining parameter estimates change between the two models.

When coefficient estimates are sensitive to which variables are included in the model, this typically indicates that some variables are collinear. For example, a possum's gender may be related to its head length, which would explain why the coefficient (and p-value) for sex male changed when we removed the head length variable. Likewise, a possum's skull width is likely to be related to its head length, probably even much more closely related than the head length was to gender.

f. Write out the form of the model. Also identify which of the following variables are positively associated (when controlling for other variables) with a possum being from Victoria: `head_l`, `skull_w`, `total_l`, and `tail_l`.

We dropped `head_l` from the model. Here is the equation:

$$\log_e\left(\frac{p_i}{1-p_i}\right) = 33.5 - 1.42 \text{ sex} - 0.28 \text{ skull width} + 0.57 \text{ total length} - 1.81 \text{ tail length}$$

Only `total_l` is positively association with the probability of being from Victoria.

g. Suppose we see a brushtail possum at a zoo in the US, and a sign says the possum had been captured in the wild in Australia, but it doesn't say which part of Australia. However, the sign does indicate that the possum is male, its skull is about 63 mm wide, its tail is 37 cm long, and its total length is 83 cm. What is the reduced model's computed probability that this possum is from Victoria? How confident are you in the model's accuracy of this probability calculation?

Let's predict the outcome. We use `response` for the type to put the answer in the form of a probability. See the help menu on `predict.glm` for more information.

```
predict(possum_mod_red,newdata = data.frame(sex="m",skull_w=63,tail_l=37,total_l=83),
        type="response",se.fit = TRUE)
```

```
## $fit
##           1
## 0.006205055
##
## $se.fit
##           1
## 0.008011468
##
## $residual.scale
## [1] 1
```

While the probability, 0.006, is very near zero, we have not run diagnostics on the model. We should also have a little skepticism that the model will hold for a possum found in a US zoo. However, it is encouraging that the possum was caught in the wild.

As a rough sense of the accuracy, we will use the standard error. The errors are really binomial but we are trying to use a normal approximation. If you remember back to our block on probability, with such a low probability, this assumption of normality is suspect. However, we will use it to give us an upper bound.

```
0.0062+c(-1,1)*1.96*.008
```

```
## [1] -0.00948  0.02188
```

So at most, the probability of the possum being from Victoria is 2%.

2. Medical school admission

The file `MedGPA.csv` in the `data` folder has information on medical school admission status and GPA and standardized test scores gathered on 55 medical school applicants from a liberal arts college in the Midwest.

The variables are:

`Accept Status`: A=accepted to medical school or D=denied admission `Acceptance`: Indicator for Accept: 1=accepted or 0=denied `Sex`: F=female or M=male `BCPM`: Bio/Chem/Physics/Math grade point average `GPA`: College grade point average `VR`: Verbal reasoning (subscore) `PS`: Physical sciences (subscore) `WS`: Writing sample (subcore) `BS`: Biological sciences (subscore) `MCAT`: Score on the MCAT exam (sum of CR+PS+WS+BS) `Apps`: Number of medical schools applied to

a. Build a logistic regression model to predict `Acceptance` from `GPA`.

```
MedGPA <- read_csv("data/MedGPA.csv")
```

```
glimpse(MedGPA)
```

```
## Rows: 55
## Columns: 11
## $ Accept     <chr> "D", "A", "A", "A", "A", "A", "A", "D", "A", "A", "A", "...
## $ Acceptance <dbl> 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0,...
## $ Sex        <chr> "F", "M", "F", "F", "F", "M", "M", "M", "F", "F", "F", "...
## $ BCPM       <dbl> 3.59, 3.75, 3.24, 3.74, 3.53, 3.59, 3.85, 3.26, 3.74, 3....
## $ GPA        <dbl> 3.62, 3.84, 3.23, 3.69, 3.38, 3.72, 3.89, 3.34, 3.71, 3....
## $ VR         <dbl> 11, 12, 9, 12, 9, 10, 11, 11, 8, 9, 11, 11, 8, 9, 11, 12...
## $ PS         <dbl> 9, 13, 10, 11, 11, 9, 12, 11, 10, 9, 9, 8, 10, 9, 8, 8, ...
## $ WS         <dbl> 9, 8, 5, 7, 4, 7, 6, 8, 6, 6, 8, 4, 7, 4, 6, 8, 8, 9, 5,...
## $ BS         <dbl> 9, 12, 9, 10, 11, 10, 11, 9, 11, 10, 11, 8, 10, 10, 7, 1...
## $ MCAT       <dbl> 38, 45, 33, 40, 35, 36, 40, 39, 35, 34, 39, 31, 35, 32, ...
## $ Apps       <dbl> 5, 3, 19, 5, 11, 5, 5, 7, 5, 11, 6, 9, 5, 8, 15, 6, 6, 1...
```

```
MedGPA <- MedGPA %>%
  mutate(Accept=factor(Accept),Sex=factor(Sex))
```

```
glimpse(MedGPA)
```

```
## Rows: 55
## Columns: 11
## $ Accept     <fct> D, A, A, A, A, A, A, D, A, A, A, A, A, D, D, A, D, A, D,...
## $ Acceptance <dbl> 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0,...
## $ Sex        <fct> F, M, F, F, F, M, M, M, F, F, F, F, M, M, M, F, M, M, M,...
## $ BCPM       <dbl> 3.59, 3.75, 3.24, 3.74, 3.53, 3.59, 3.85, 3.26, 3.74, 3....
## $ GPA        <dbl> 3.62, 3.84, 3.23, 3.69, 3.38, 3.72, 3.89, 3.34, 3.71, 3....
## $ VR         <dbl> 11, 12, 9, 12, 9, 10, 11, 11, 8, 9, 11, 11, 8, 9, 11, 12...
## $ PS         <dbl> 9, 13, 10, 11, 11, 9, 12, 11, 10, 9, 9, 8, 10, 9, 8, 8, ...
## $ WS         <dbl> 9, 8, 5, 7, 4, 7, 6, 8, 6, 6, 8, 4, 7, 4, 6, 8, 8, 9, 5,...
## $ BS         <dbl> 9, 12, 9, 10, 11, 10, 11, 9, 11, 10, 11, 8, 10, 10, 7, 1...
## $ MCAT       <dbl> 38, 45, 33, 40, 35, 36, 40, 39, 35, 34, 39, 31, 35, 32, ...
## $ Apps       <dbl> 5, 3, 19, 5, 11, 5, 5, 7, 5, 11, 6, 9, 5, 8, 15, 6, 6, 1...
```

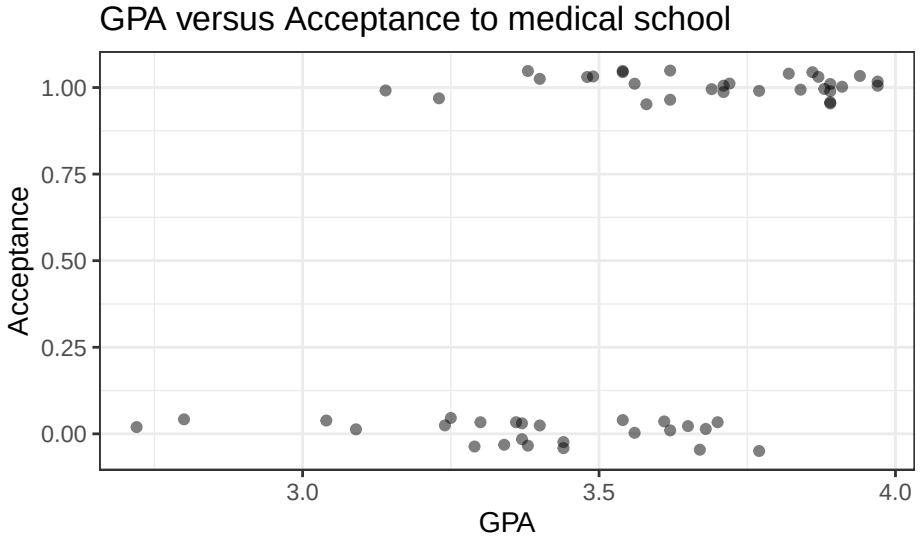
```
med_mod<-glm(Acceptance~GPA,data=MedGPA,family=binomial)
```

```
summary(med_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
```

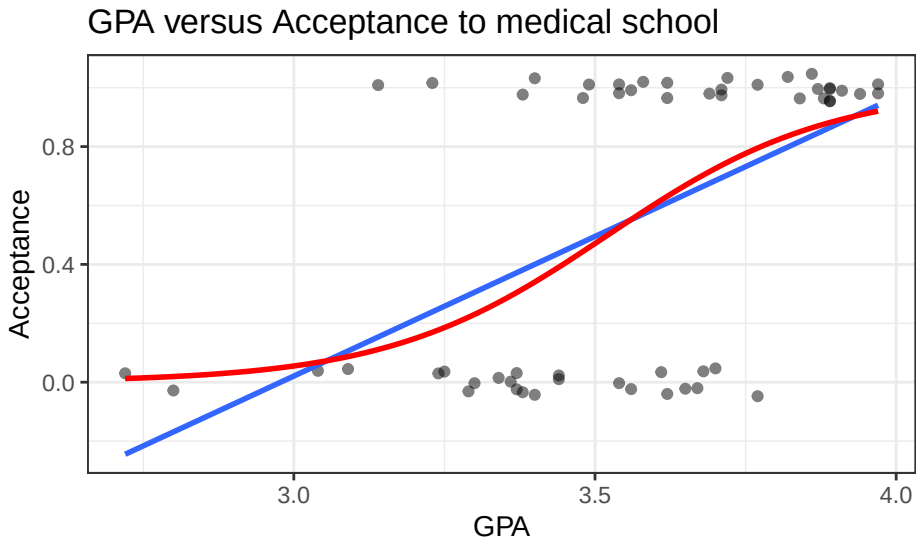
b. Plot `Acceptance` versus `GPA`, add *jitter* in the vertical direction.

```
ggplot(data = MedGPA, aes(x = GPA, y = Acceptance)) +
  geom_jitter(width = 0, height = 0.05, alpha = 0.5) +
  labs(title="GPA versus Acceptance to medical school") +
  theme_bw()
```



c. Repeat the plot in part b but add linear and logistic fitted line to the plot.

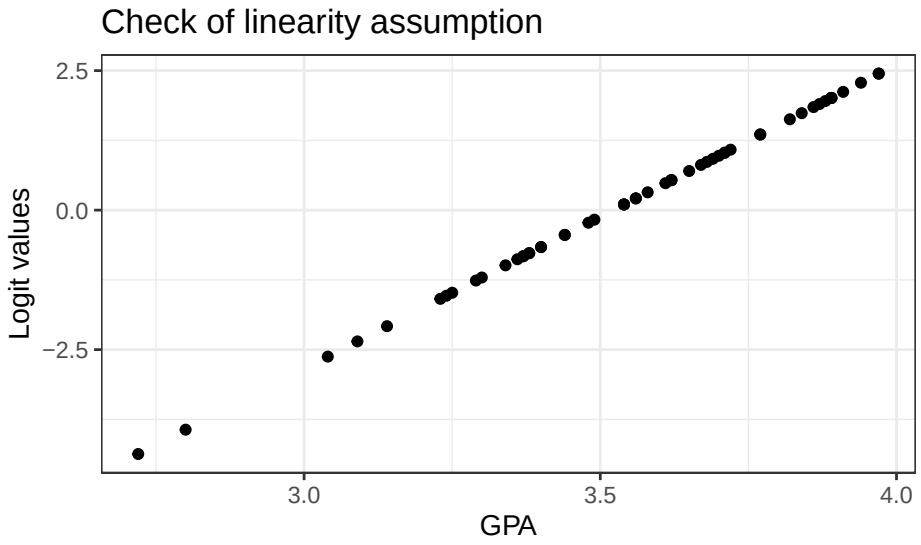
```
ggplot(data = MedGPA, aes(x = GPA, y = Acceptance)) +
  geom_jitter(width = 0, height = 0.05, alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +   geom_smooth(method = "glm", se = FALSE, color = "red",
  labs(title="GPA versus Acceptance to medical school") +
  theme_bw()
```



d. Check the linearity assumption by plotting `GPA` versus the logit of `Acceptance`, the response on the logit scale.

We will use `augment()` to help us.

```
ggplot(augment(med_mod),aes(x=GPA,y=.fitted)) +
  geom_point() +
  theme_bw() +
  labs(title="Check of linearity assumption",
       y="Logit values")
```

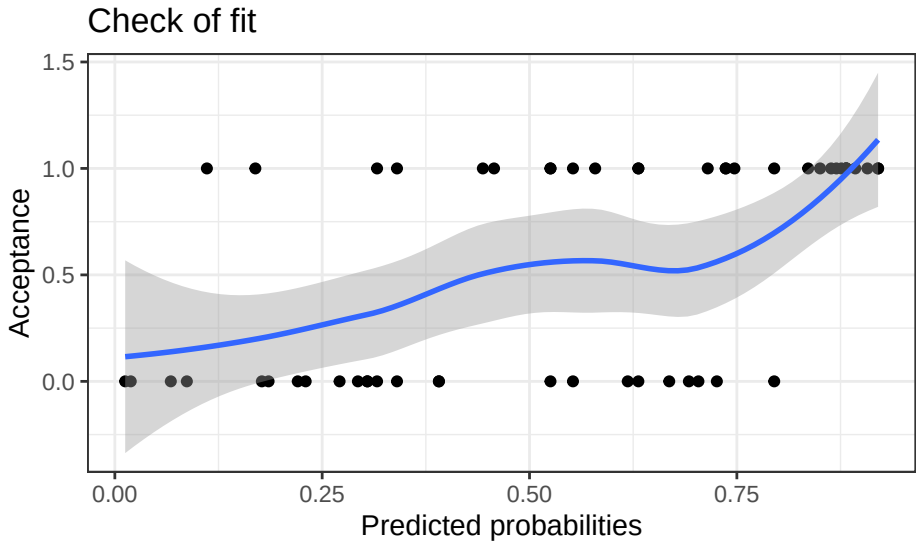


We have too few data points to make this plot useful. We need to be able to bin the data points together to create an n greater than 1. See https://online.stat.psu.edu/stat504/node/160/ for more information.

If the model fits well a smooth fit between the predicted probabilities and actual values should be close to linear. In the following plot, we may not hae a good fit.

```
ggplot(augment(med_mod,type.predict = "response"),aes(x=.fitted,y=Acceptance)) +
  geom_point()+
  geom_smooth(method="loess") +
  theme_bw() +
  labs(title="Check of fit",
       x="Predicted probabilities")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



File Creation Information

- File creation date: 2020-12-11
- Windows version: Windows 10 x64 (build 18362)
- R version 3.6.3 (2020-02-29)
- `mosaic` package version: 1.7.0
- `tidyverse` package version: 1.3.0
- `openintro` package version: 2.0.0