Multiple and Logistic Regression in R_Logistic Regression

dizhen

5/6/2020

What is logistic regression?

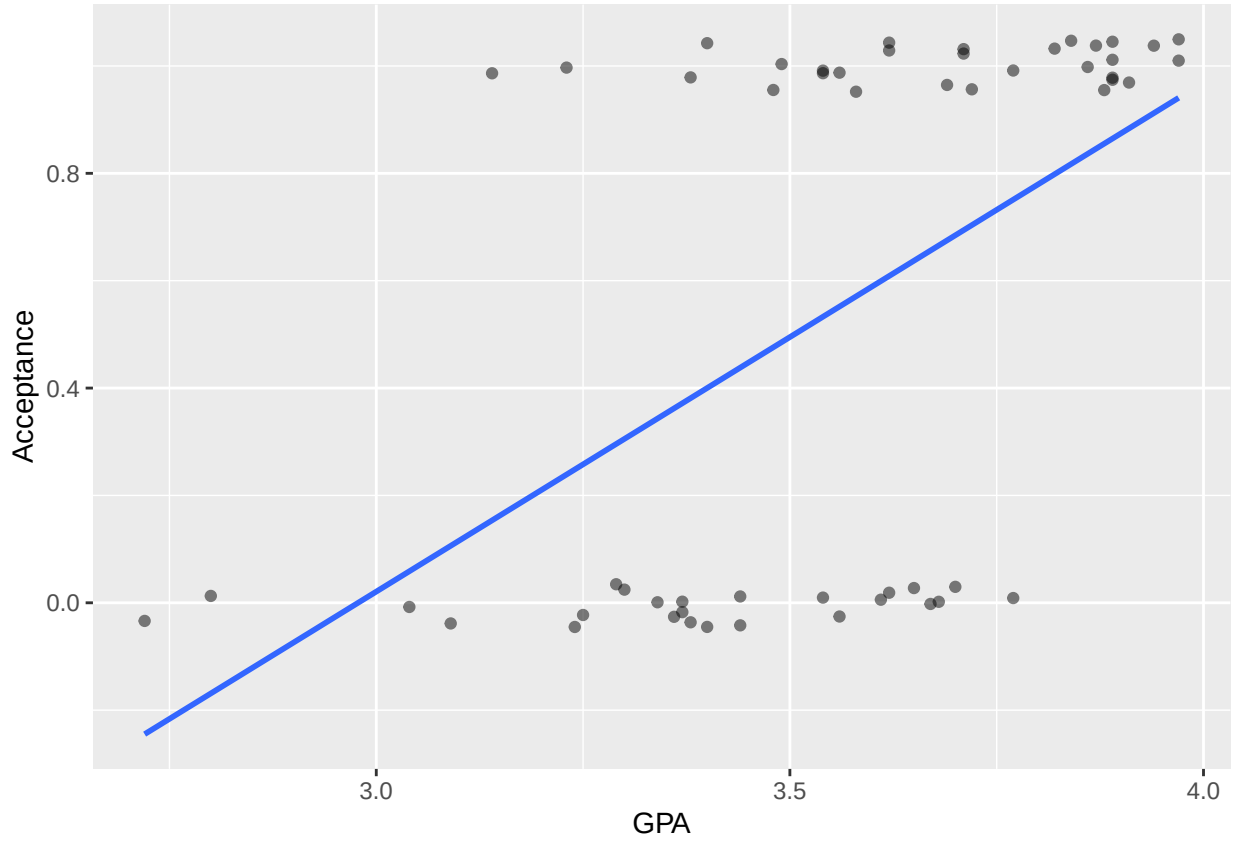
```
library(Stat2Data)
library(ggplot2)
library(dplyr)
```

```
##
## 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
```

```
library(broom)
data(MedGPA)
```

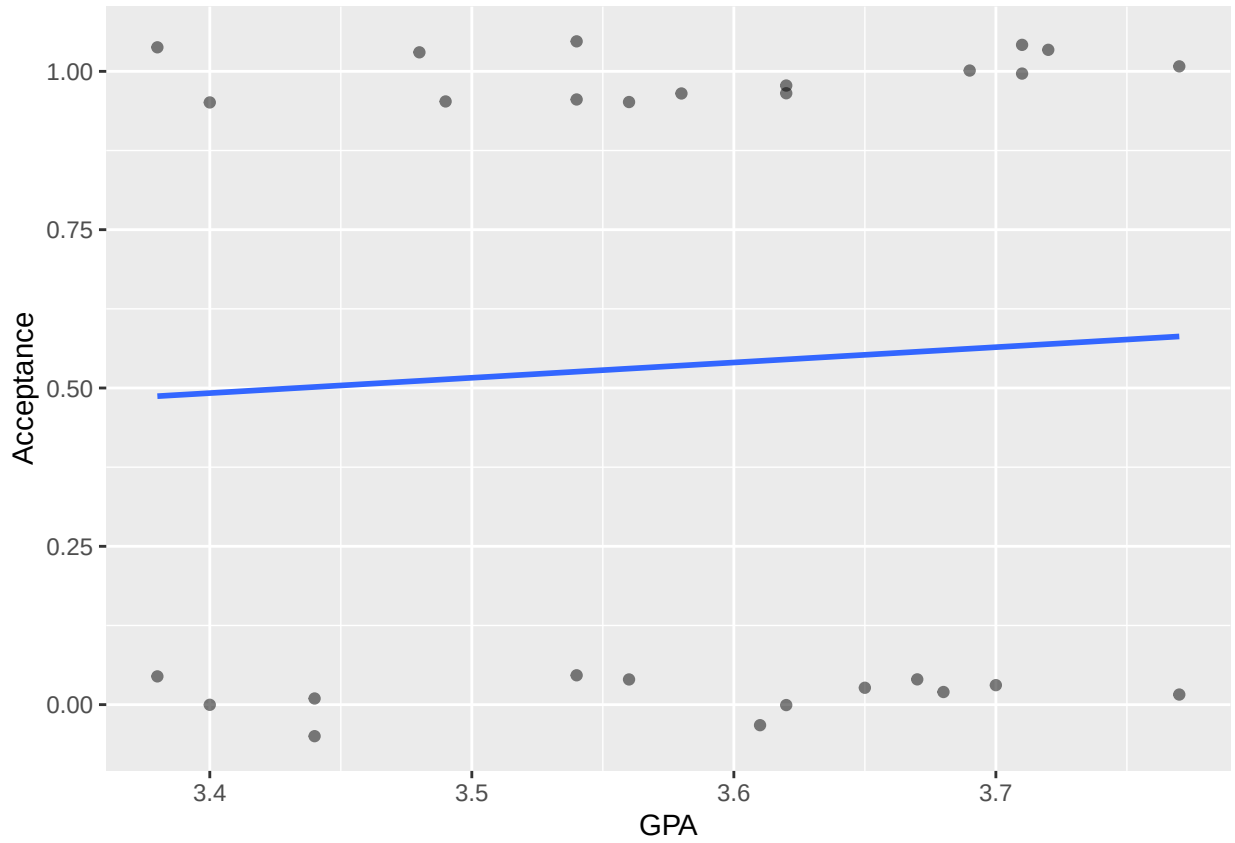
```
# create a scatterplot for acceptance as a function of GPA (y as a function of x)
# scatterplot with jitter
data_space <- ggplot(data = MedGPA, aes(y = Acceptance, x = GPA)) +
  geom_jitter(width = 0, height = 0.05, alpha = 0.5)

# linear regression line
data_space +
  geom_smooth(method = "lm", se = FALSE)
```

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




```
# filter
MedGPA_middle <- MedGPA %>%
  filter(GPA >= 3.375, GPA <= 3.770)

# scatterplot with jitter
data_space <- ggplot(data = MedGPA_middle, aes(y = Acceptance, x = GPA)) +
  geom_jitter(width = 0, height = 0.05, alpha = 0.5)

# linear regression line
data_space +
  geom_smooth(method = "lm", se = FALSE)
```

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




```
# fit model
glm(Acceptance ~ GPA, data = MedGPA, family = binomial)
```

```
##
## Call:  glm(formula = Acceptance ~ GPA, family = binomial, data = MedGPA)
##
## Coefficients:
## (Intercept)          GPA
##     -19.207        5.454
##
## Degrees of Freedom: 54 Total (i.e. Null);  53 Residual
## Null Deviance:       75.79
## Residual Deviance: 56.84     AIC: 60.84
```

Visualizing logistic regression

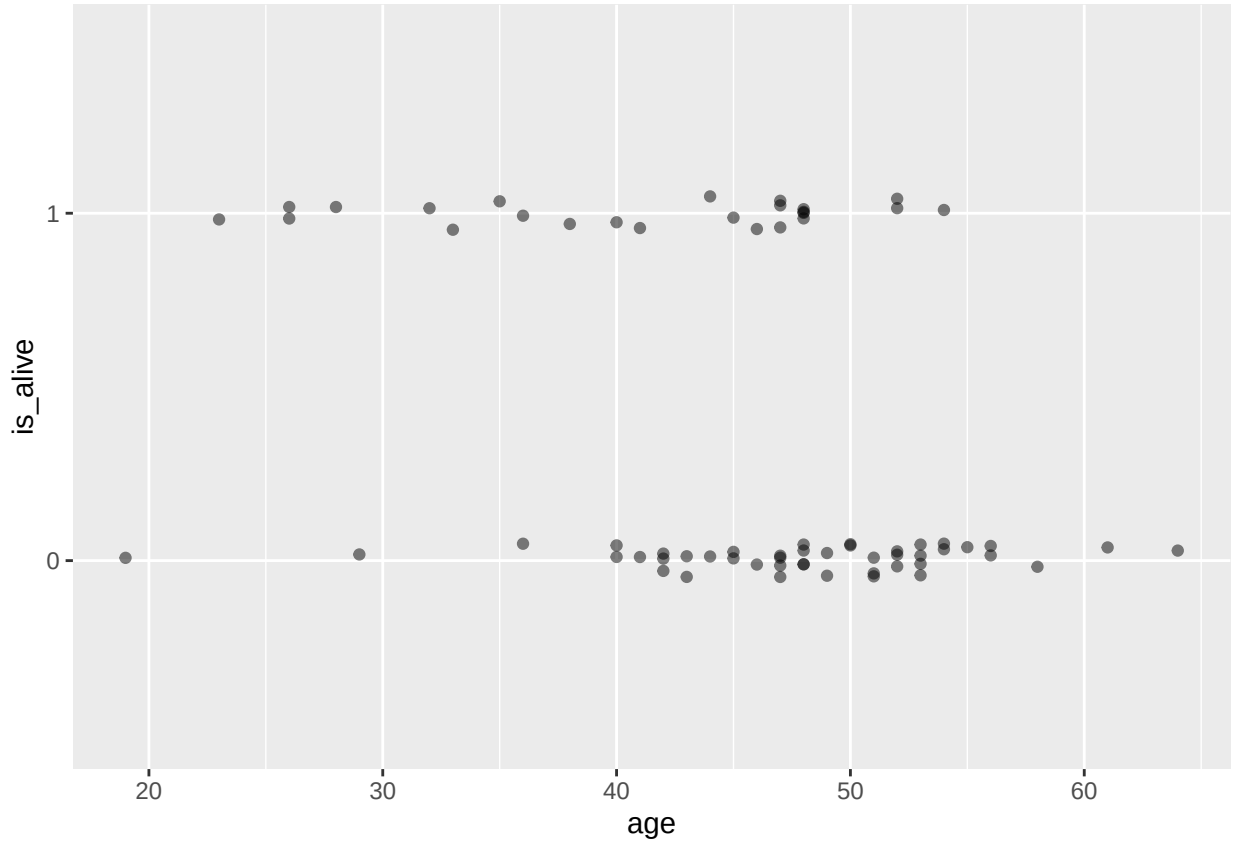
Visualizing a binary response

```
library(openintro)
```

```
## Please visit openintro.org for free statistics materials

##
## Attaching package: 'openintro'
```

```
## The following object is masked from 'package:ggplot2':
##
##     diamonds

## The following objects are masked from 'package:datasets':
##
##     cars, trees
```

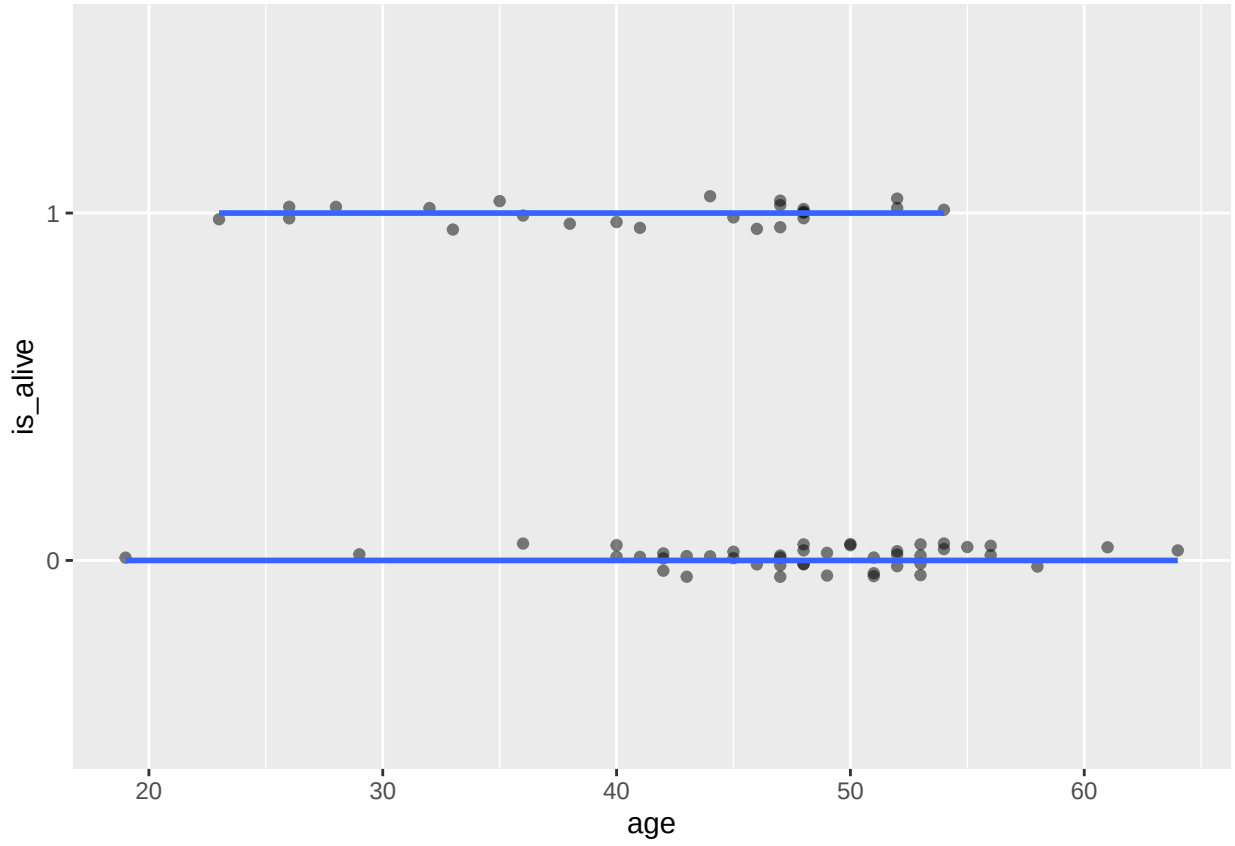
```
data("heartTr")
heartTr <- heartTr %>%
  mutate(is_alive = ifelse(survived == "dead", 0,1)) %>%
  mutate(is_alive = as.factor(is_alive)) %>%
  na.omit()

data_space <- ggplot(data = heartTr, aes(x = age, y = is_alive)) +
  geom_jitter(width = 0, height = 0.05, alpha = 0.5)
data_space
```



Regression with a binary response

```
data_space +
  geom_smooth(method = "lm", se = FALSE)
```

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



Fitting a GLM

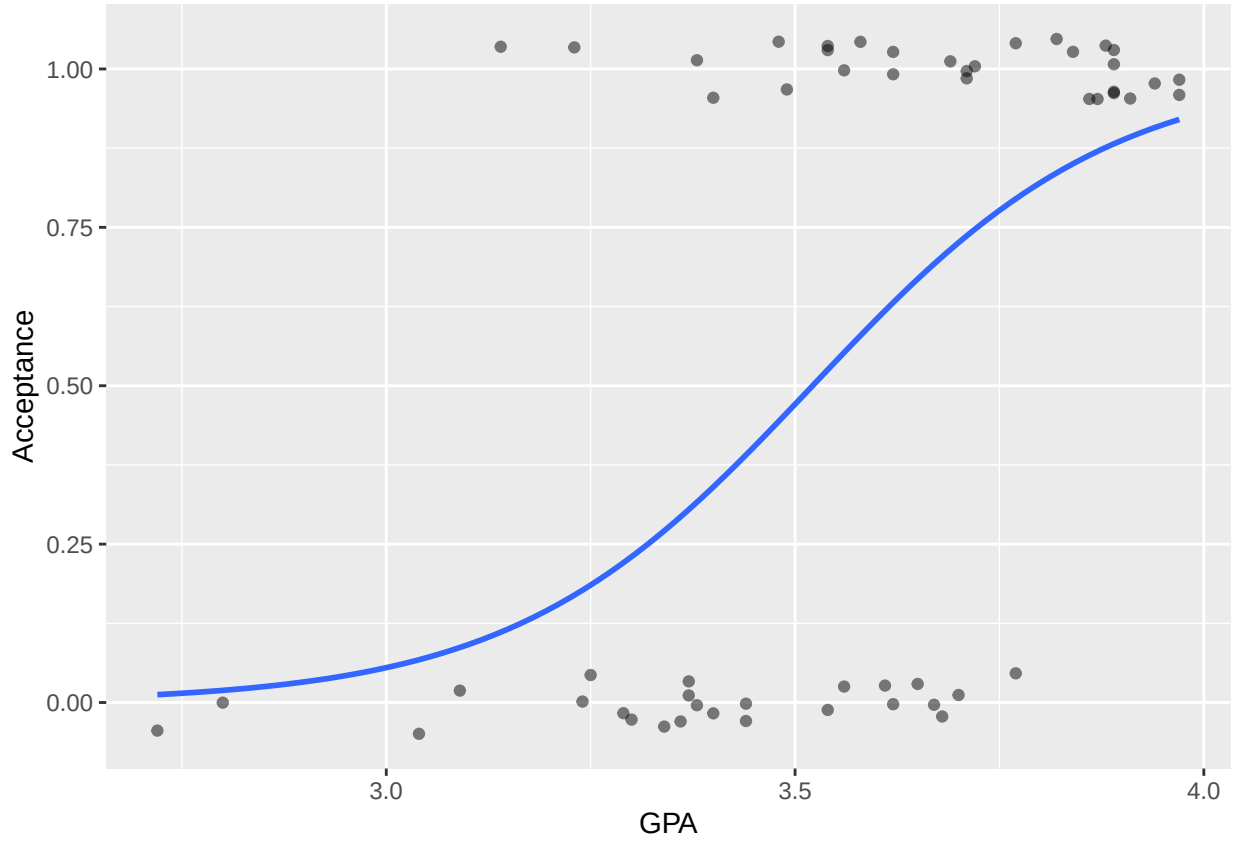
```
mod <- glm(is_alive ~ age, data = heartTr, family = binomial)
mod
```

```
##
## Call:  glm(formula = is_alive ~ age, family = binomial, data = heartTr)
##
## Coefficients:
## (Intercept)          age
##     3.67591     -0.09605
##
## Degrees of Freedom: 68 Total (i.e. Null);  67 Residual
## Null Deviance:       89.16
## Residual Deviance: 79.43     AIC: 83.43
```

Practice

```
# scatterplot with jitter
data_space <- ggplot(data = MedGPA, aes(y = Acceptance, x = GPA)) +
  geom_jitter(width = 0, height = 0.05, alpha = 0.5)

# add logistic curve
data_space +
  geom_smooth(method = "glm", se = FALSE, method.args = list(family = "binomial"))
```

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




```
gpa_bins <- quantile(MedGPA$GPA, probs = seq(0, 1, 1/6))
gpa_bins
```

```
##        0% 16.66667% 33.33333%       50% 66.66667% 83.33333%      100%
##      2.72      3.30      3.44      3.58      3.70      3.87      3.97
```

```
MedGPA$bins <- cut(MedGPA$GPA, breaks = gpa_bins, include.lowest = TRUE)
head(MedGPA)
```

```
##   Accept Acceptance Sex BCPM  GPA VR PS WS BS MCAT Apps       bins
## 1      D          0   F 3.59 3.62 11  9  9  9   38    5 (3.58,3.7]
## 2      A          1   M 3.75 3.84 12 13  8 12   45    3 (3.7,3.87]
## 3      A          1   F 3.24 3.23  9 10  5  9   33   19 [2.72,3.3]
## 4      A          1   F 3.74 3.69 12 11  7 10   40    5 (3.58,3.7]
## 5      A          1   F 3.53 3.38  9 11  4 11   35   11 (3.3,3.44]
## 6      A          1   M 3.59 3.72 10  9  7 10   36    5 (3.7,3.87]
```

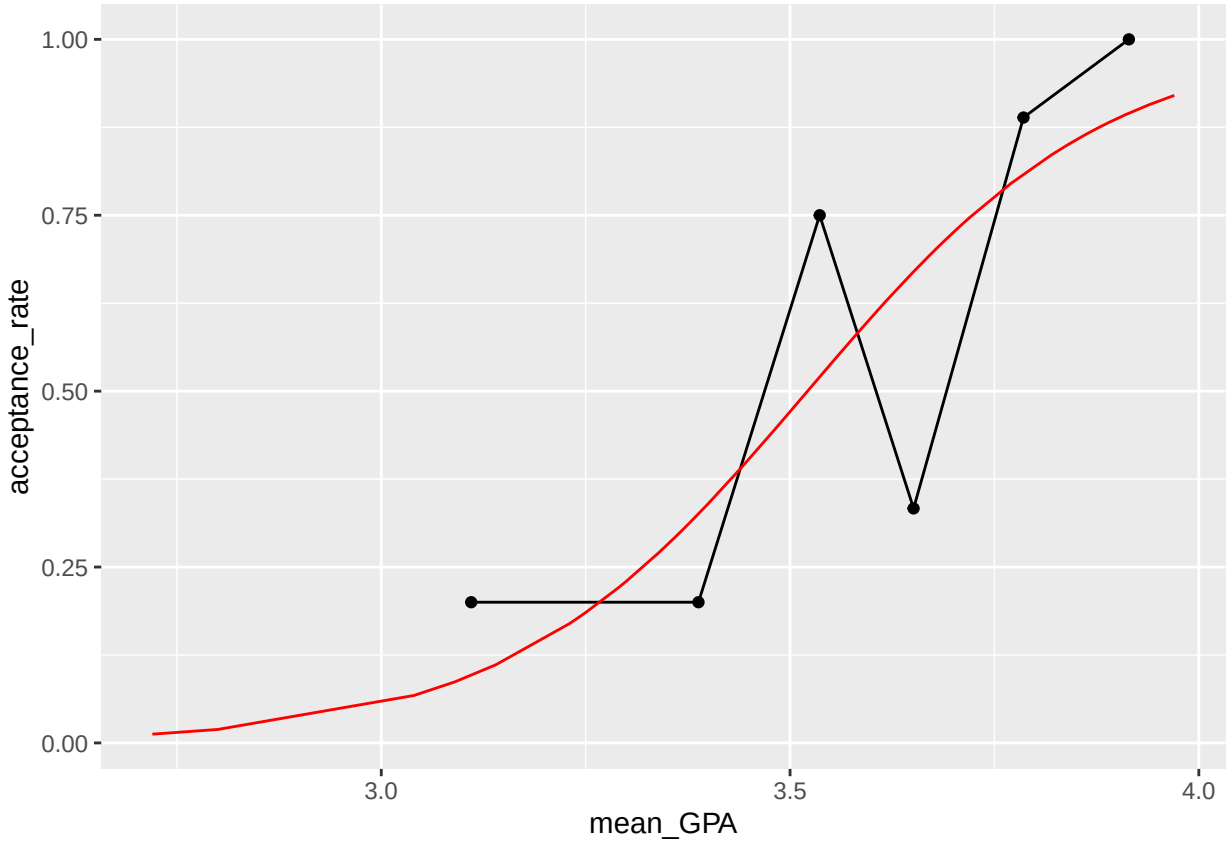
```
MedGPA_binned <- MedGPA %>%
  group_by(bins) %>%
  summarize(mean_GPA = mean(GPA), acceptance_rate = mean(Acceptance))
MedGPA_binned
```

```
## # A tibble: 6 x 3
##   bins         mean_GPA acceptance_rate
##   <fct>           <dbl>           <dbl>
## 1 [2.72,3.3]       3.11           0.2
## 2 (3.3,3.44]       3.39           0.2
## 3 (3.44,3.58]      3.54           0.75
## 4 (3.58,3.7]       3.65           0.333
## 5 (3.7,3.87]       3.79           0.889
## 6 (3.87,3.97]      3.91           1
```

```
library(broom)
# fit model
mod <- glm(Acceptance ~ GPA, data = MedGPA, family = binomial)
mod
```

```
##
## Call:  glm(formula = Acceptance ~ GPA, family = binomial, data = MedGPA)
##
## Coefficients:
## (Intercept)          GPA
##     -19.207        5.454
##
## Degrees of Freedom: 54 Total (i.e. Null);  53 Residual
## Null Deviance:       75.79
## Residual Deviance: 56.84     AIC: 60.84
```

```
# binned points and line
data_space <- ggplot(data = MedGPA_binned, aes(x = mean_GPA, y = acceptance_rate)) +
  geom_point() + geom_line()

# augmented model
MedGPA_plus <- mod %>%
  augment(type.predict = "response")

# logistic model on probability scale
data_space +
  geom_line(data = MedGPA_plus, aes(x = GPA, y = .fitted), color = "red")
```



Three scales approach to interpretation

1. Probability scale

```
mod <- glm(is_alive ~ age, data = heartTr, family = binomial)
mod
```

```
##
## Call:  glm(formula = is_alive ~ age, family = binomial, data = heartTr)
##
## Coefficients:
## (Intercept)          age
##     3.67591     -0.09605
##
## Degrees of Freedom: 68 Total (i.e. Null);  67 Residual
## Null Deviance:       89.16
## Residual Deviance: 79.43     AIC: 83.43
```

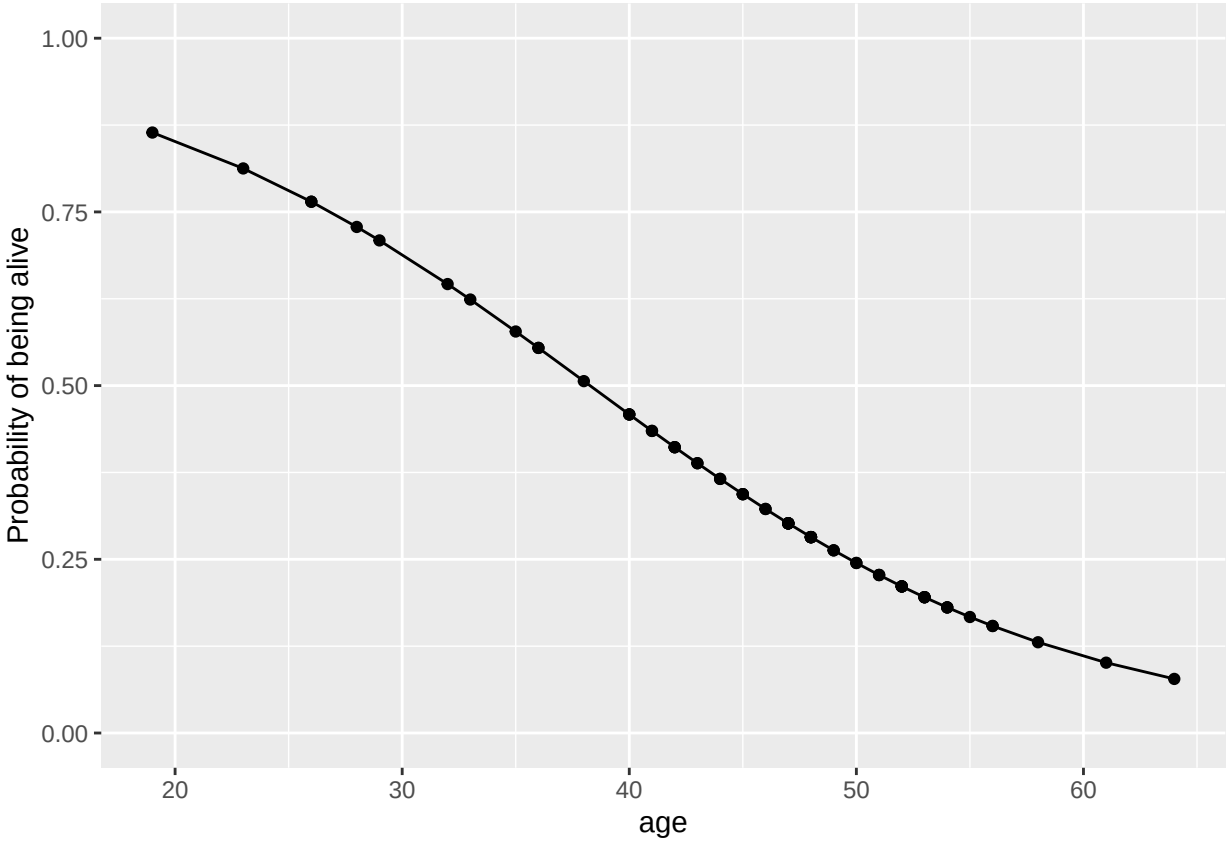
```
heartTr_plus <- mod %>%
  augment(type.predict = "response") %>%
  mutate(y_hat = .fitted);heartTr_plus
```

```
## # A tibble: 69 x 11
##    .rownames is_alive   age .fitted .se.fit .resid   .hat .sigma .cooksd
```

```
##    <chr>     <fct>    <int>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>   <dbl>
##  1 8         0           41   0.435  0.0734 -1.07  0.0220   1.09 0.00883
##  2 16        0           40   0.459  0.0781 -1.11  0.0245   1.09 0.0109
##  3 18        0           54   0.181  0.0623 -0.632 0.0262   1.09 0.00305
##  4 19        0           29   0.709  0.123  -1.57  0.0736   1.08 0.104
##  5 23        0           55   0.167  0.0622 -0.605 0.0278   1.09 0.00295
##  6 24        0           52   0.211  0.0618 -0.688 0.0230   1.09 0.00321
##  7 30        0           40   0.459  0.0781 -1.11  0.0245   1.09 0.0109
##  8 31        1           35   0.578  0.104   1.05  0.0442   1.09 0.0177
##  9 34        0           56   0.154  0.0618 -0.578 0.0293   1.09 0.00284
## 10 35        0           36   0.554  0.0989 -1.27  0.0396   1.09 0.0267
## # ... with 59 more rows, and 2 more variables: .std.resid <dbl>, y_hat <dbl>
```

Probability scale plot

```
ggplot(heartTr_plus, aes(x = age, y = y_hat)) +
  geom_point() + geom_line() +
  scale_y_continuous("Probability of being alive", limits = c(0, 1))
```

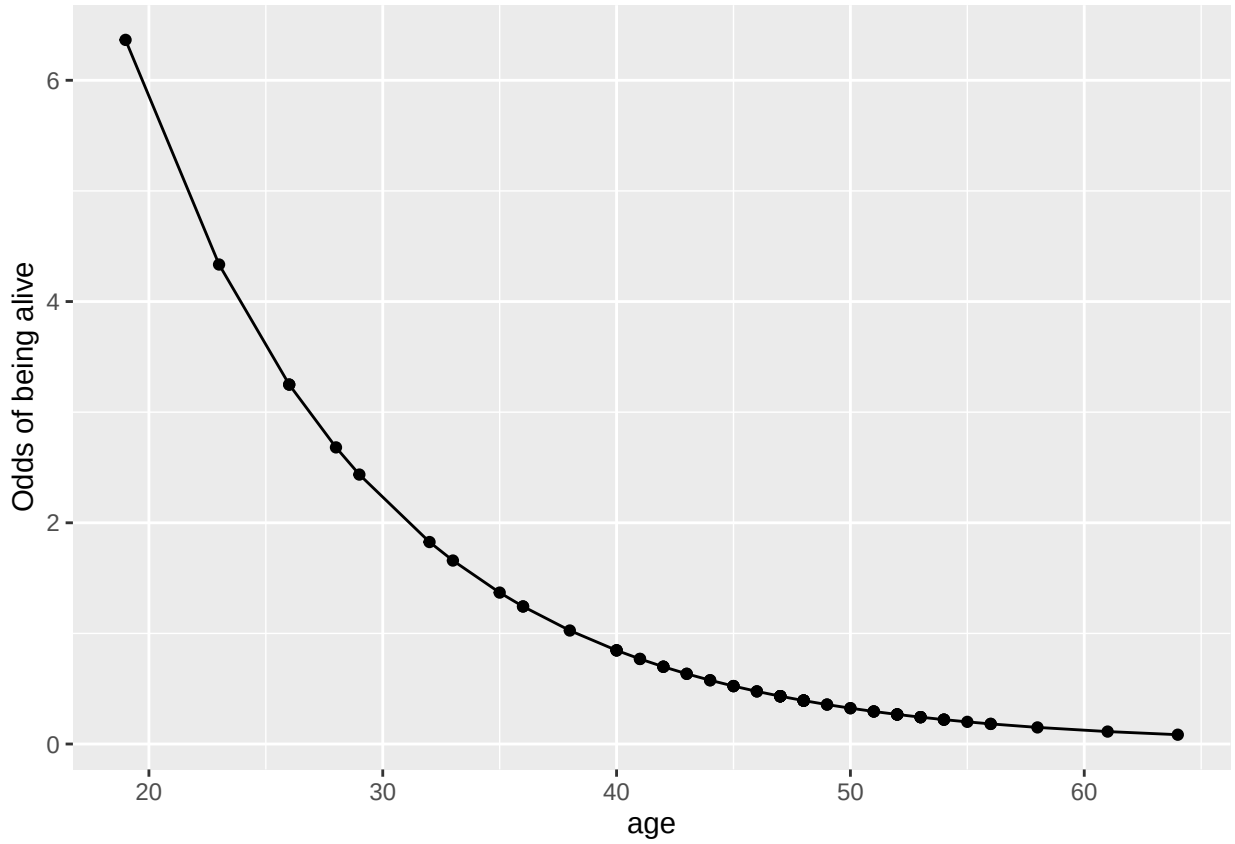


2. Odds scale

```
heartTr_plus <- heartTr_plus %>%
  mutate(odds_hat = y_hat / (1 - y_hat))
```

Odds scale plot

```
ggplot(heartTr_plus, aes(x = age, y = odds_hat)) +
  geom_point() + geom_line() +
  scale_y_continuous("Odds of being alive")
```

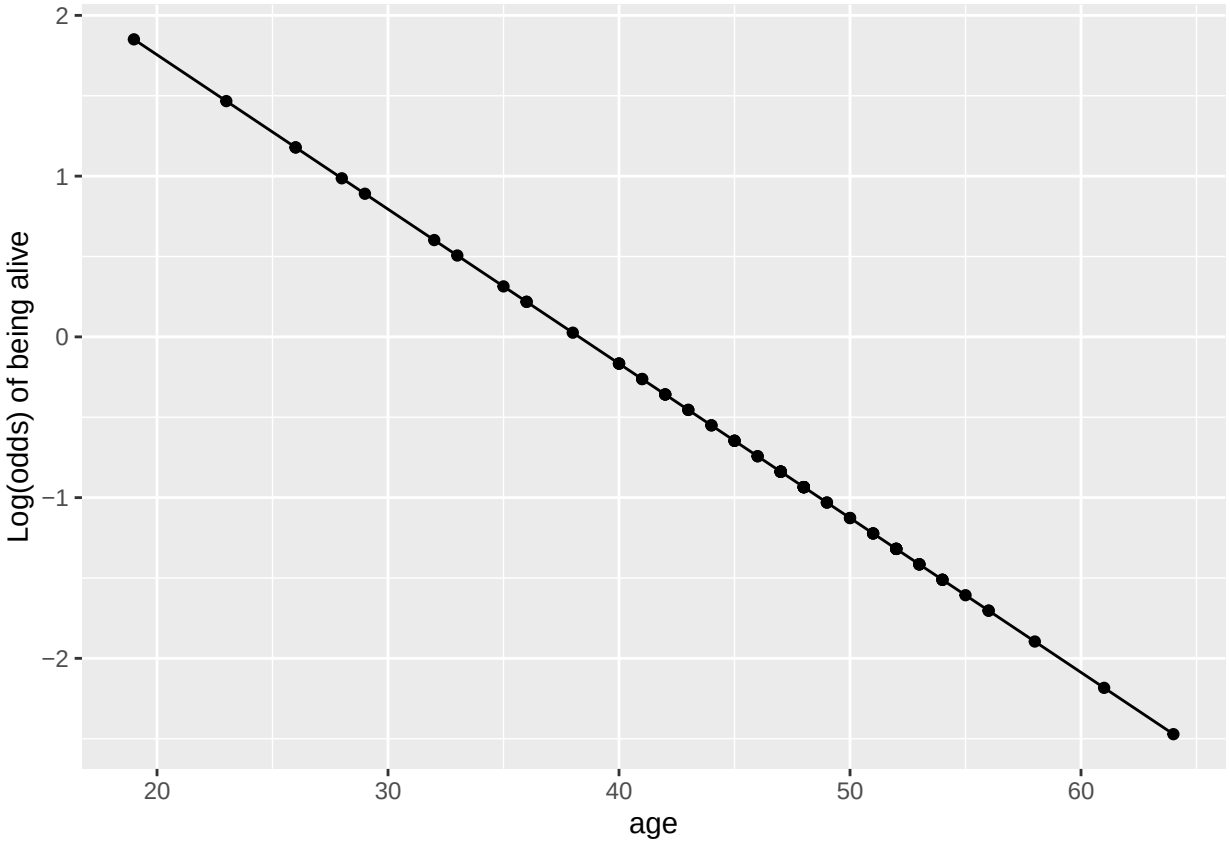


3. Log-odds scale

```
heartTr_plus <- heartTr_plus %>%
  mutate(log_odds_hat = log(odds_hat))
```

Log-odds plot

```
ggplot(heartTr_plus, aes(x = age, y = log_odds_hat)) +
  geom_point() + geom_line() +
  scale_y_continuous("Log(odds) of being alive")
```



Comparison

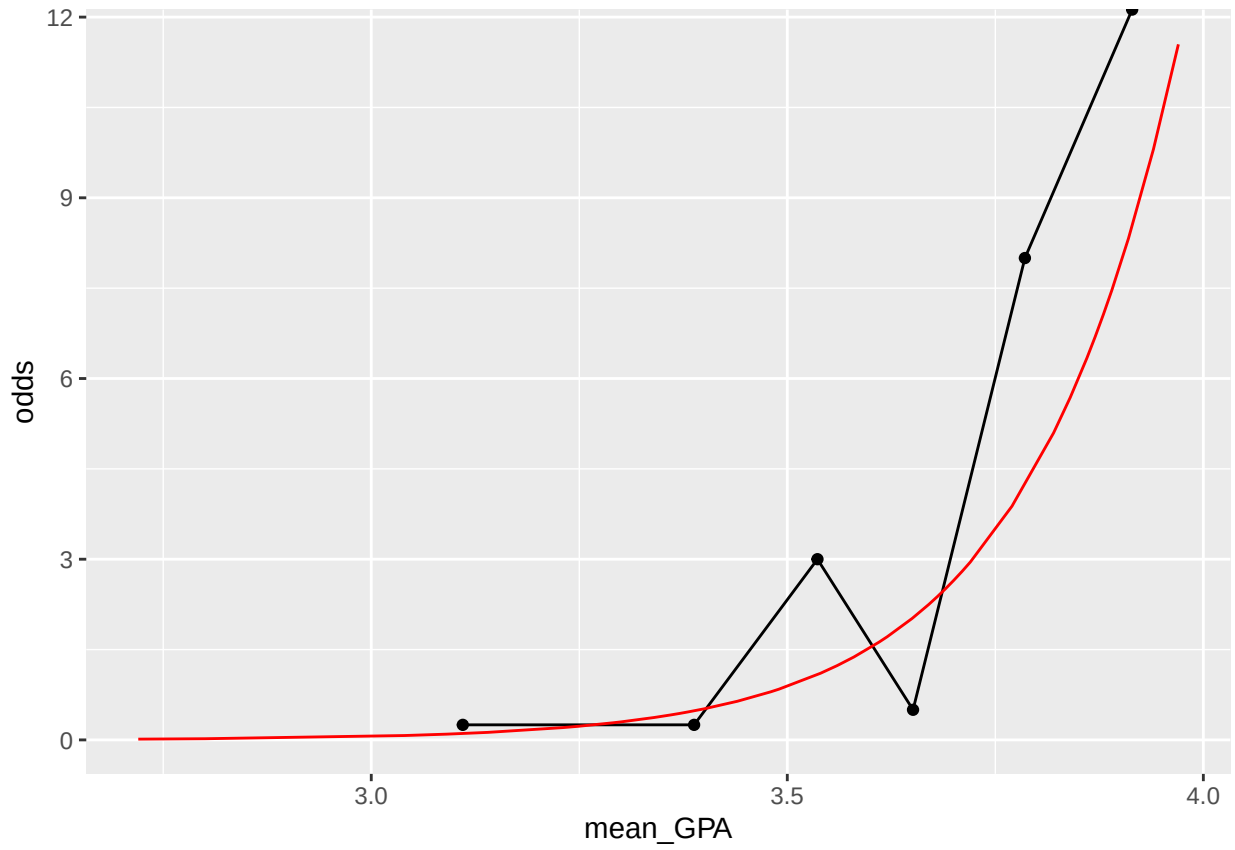
- Probability scale
  - scale: intuitive, easy to interpret
  - function: non-linear, hard to interpret
- Odds scale
  - scale: harder to interpret
  - function: exponential, harder to interpret
- Log-odds scale
  - scale: impossible to interpret
  - function: linear, easy to interpret

Practice

If the probability of getting accepted is y, then the odds are y/(1-y).

```
# compute odds for bins
MedGPA_binned <- MedGPA_binned %>%
  mutate(odds = acceptance_rate / (1 - acceptance_rate))

# plot binned odds
data_space <- ggplot(data = MedGPA_binned, aes(x = mean_GPA, y = odds)) +
  geom_point() + geom_line()
```

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

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

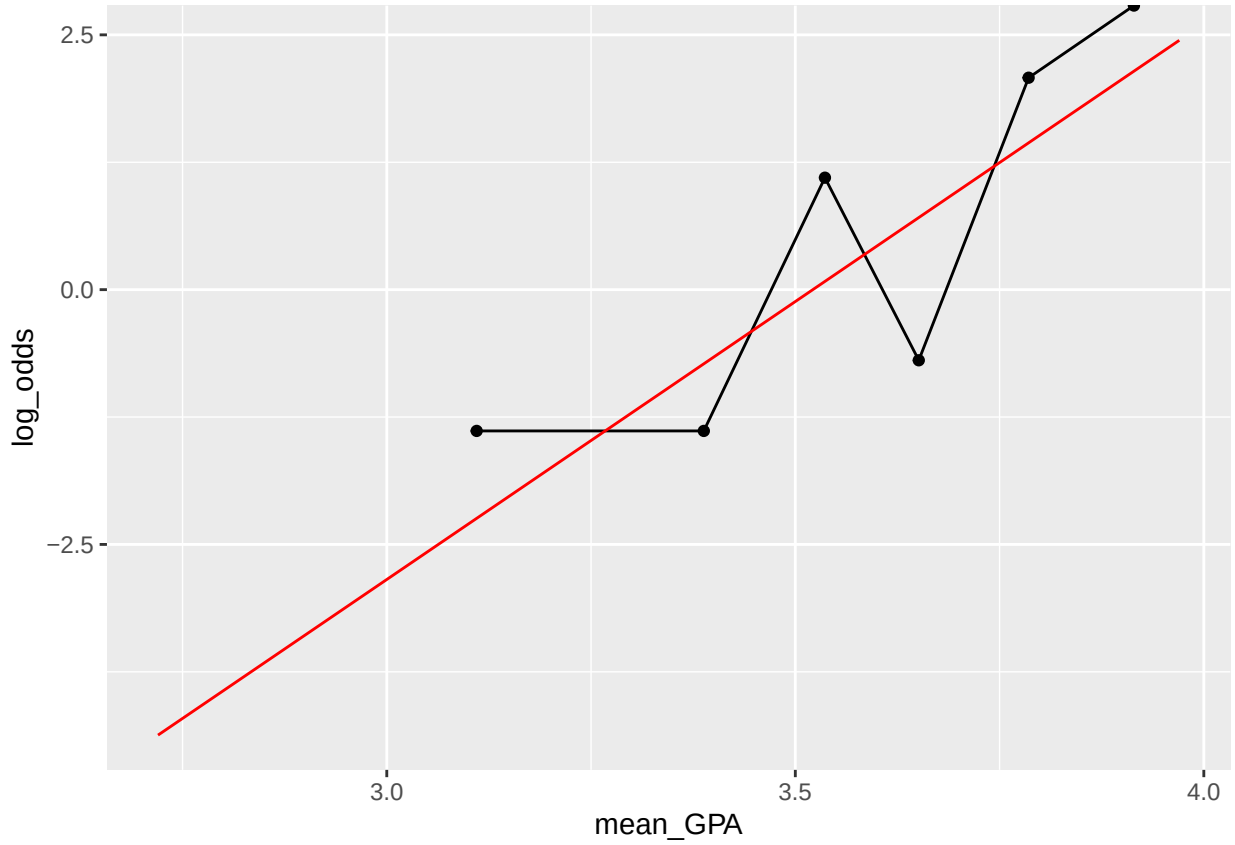


On the log-odds scale, the units are nearly impossible to interpret, but the function is linear, which makes it easy to understand

None of these three is uniformly superior.The interpretation of the coefficients is most commonly done on the odds scale. On the odds scale, a one unit change in x leads to the odds being multiplied by a factor of β_1.

```r
# compute log odds for bins
MedGPA_binned <- MedGPA_binned %>%
  mutate(log_odds = log(acceptance_rate / (1 - acceptance_rate)))

# plot binned log odds
data_space <- 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)))
```

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



Using a logistic model

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

# make predictions
augment(mod, newdata = new_data, type.predict = "response")
```

```
## # A tibble: 69 x 2
##      GPA .fitted
##    <dbl>   <dbl>
##  1  3.51   0.435
##  2  3.51   0.459
##  3  3.51   0.181
##  4  3.51   0.709
##  5  3.51   0.167
##  6  3.51   0.211
##  7  3.51   0.459
```

```
##  8  3.51   0.578
##  9  3.51   0.154
## 10  3.51   0.554
## # ... with 59 more rows
```

```
# fit model
mod <- glm(Acceptance ~ GPA, data = MedGPA, family = binomial)
mod
```

```
##
## Call:  glm(formula = Acceptance ~ GPA, family = binomial, data = MedGPA)
##
## Coefficients:
## (Intercept)          GPA
##     -19.207        5.454
##
## Degrees of Freedom: 54 Total (i.e. Null);  53 Residual
## Null Deviance:       75.79
## Residual Deviance: 56.84     AIC: 60.84
```

```
# 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
```