Logistic Regression with the Lindner Data

Thomas E. Love

3/6/2020

```
library(here); library(janitor); library(magrittr)
library(knitr)
library(rms)
library(caret)
library(ROCR)
library(pROC)
library(broom)
library(tidyverse)

theme_set(theme_bw())
```

```
lind <- readRDS(here("data", "lind.Rds"))

str(lind)
```

```
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':    970 obs. of  8 variables:
 $ ptid    : chr  "1001" "1002" "1003" "1004" ...
 $ cardbill: int  3563 4694 7366 8247 8319 8410 8517 8763 8823 8970 ...
 $ abcix   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ stent   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ acutemi : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ejecfrac: int  56 50 50 55 50 58 30 60 60 60 ...
 $ ves1proc: int  1 1 1 1 1 1 1 1 1 1 ...
 $ diabetic: int  0 0 1 0 0 0 0 0 0 0 ...
```

- Note that the `abcix` variable is represented as an integer, with possible values 0 and 1, as is `stent`.

```
lind %>% count(abcix, stent)
```

```
# A tibble: 4 x 3
  abcix stent     n
  <int> <int> <int>
1     0     0   118
2     0     1   165
3     1     0   203
4     1     1   484
```

- The `ejecfrac` is a quantitative variable, and the units for `ejecfrac` are percentage points. All of the values are integers, and we observe 29 distinct values.

```
lind %$% Hmisc::describe(ejecfrac)
```

```
ejecfrac
       n  missing distinct     Info     Mean      Gmd      .05      .10
     970        0       29    0.976    51.18    10.49       30       40
```

```
     .25      .50      .75      .90      .95
      45       55       56       60       60

lowest :  0 15 19 20 25, highest: 69 70 75 80 90
```

- The `ves1proc` is represented here as an integer, but only has 6 possible values (0, 1, 2, 3, 4 and 5), since it is really a count.

```
lind %>% tabyl(ves1proc)
```

```
 ves1proc   n     percent
        0   4 0.004123711
        1 661 0.681443299
        2 249 0.256701031
        3  41 0.042268041
        4  14 0.014432990
        5   1 0.001030928
```

- I want to have a multi-categorical factor for the demonstrations that follow, so I will create a factor version of `ves1proc`, and place it in a new variable called `ves_f`. I'm not implying this is a great idea for these data. As we'll see, it leads to disaster.

```
lind_new <- lind %>%
    mutate(ves_f = factor(ves1proc))
```

`model_1`: What do I mean by a logistic regression model exploding?

Let's run a logistic regression model to predict `abcix`, a numeric (1/0) variable on the basis of `stent`, `ejecfrac` and `ves_f`.

```
model_1 <- lind_new %$%
    glm(abcix ~ stent + ejecfrac + ves_f,
        family = binomial)

summary(model_1)
```

```

Call:
glm(formula = abcix ~ stent + ejecfrac + ves_f, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3076  -1.2448   0.6510   0.8795   1.2461

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.43469    1.22536   1.987 0.046931 *
stent         0.59332    0.15191   3.906 9.39e-05 ***
ejecfrac     -0.02601    0.00790  -3.293 0.000991 ***
ves_f1       -0.84696    1.16294  -0.728 0.466439
ves_f2       -0.02341    1.17136  -0.020 0.984056
ves_f3        0.77859    1.27354   0.611 0.540963
ves_f4        1.32605    1.55694   0.852 0.394379
ves_f5       12.43202  535.41242   0.023 0.981475
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1171.2  on 969  degrees of freedom
Residual deviance: 1106.7  on 962  degrees of freedom
AIC: 1122.7

Number of Fisher Scoring iterations: 12
```

Notice the very, very large estimate for the `ves_f5` coefficient, and the huge standard error? What's the explanation?

```
lind_new %>% tabyl(abcix, ves_f)
```

```
 abcix 0   1   2  3  4 5
     0 1 230  47  4  1 0
     1 3 431 202 37 13 1
```

Some of the levels of our `ves_f` variable are very, very small, and one of them (5) is so small that we have zero subjects with `abcix` = 0 and `ves_f` = 5.

Collapsing the `ves1proc` to a factor with three levels

So, what should we do? we'll collapse the `ves_f` factor to just three levels, so we don't have any really tiny sample sizes.

```
lind_new <- lind %>%
    mutate(ves_f = factor(ves1proc),
           ves_fix = fct_recode(ves_f,
                                "Low" = "0",
                                "Low" = "1",
                                "Mid" = "2",
                                "High" = "3",
                                "High" = "4",
                                "High" = "5"))

# sanity check

lind_new %>% tabyl(ves_f, ves_fix)
```

```
 ves_f Low Mid High
     0   4   0    0
     1 661   0    0
     2   0 249    0
     3   0   0   41
     4   0   0   14
     5   0   0    1
```

```
# check that abcix can be 1 or 0 at each level of new factor ves_fix

lind_new %>% tabyl(abcix, ves_fix)
```

```
 abcix Low Mid High
     0 231  47    5
     1 434 202   51
```

`model_2`: A main effects model

OK. Let's try again, now.

```
model_2 <- lind_new %$%
    glm(abcix ~ stent + ejecfrac + ves_fix,
        family = binomial)

summary(model_2)
```

```

Call:
glm(formula = abcix ~ stent + ejecfrac + ves_fix, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2825  -1.2507   0.6522   0.8793   1.2445

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.596272   0.429099   3.720 0.000199 ***
stent        0.580387   0.151295   3.836 0.000125 ***
ejecfrac    -0.025922   0.007899  -3.282 0.001032 **
ves_fixMid   0.818329   0.183391   4.462 8.11e-06 ***
ves_fixHigh  1.777070   0.479854   3.703 0.000213 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1171.2  on 969  degrees of freedom
Residual deviance: 1107.8  on 965  degrees of freedom
AIC: 1117.8

Number of Fisher Scoring iterations: 4
```

```
tidy(model_2, exponentiate = TRUE, conf.int = TRUE) %>%
    select(term, estimate, conf.low, conf.high) %>%
    kable(digits = 3)
```

term	estimate	conf.low	conf.high
(Intercept)	4.935	2.168	11.685
stent	1.787	1.328	2.404
ejecfrac	0.974	0.959	0.989
ves_fixMid	2.267	1.594	3.275
ves_fixHigh	5.913	2.533	17.295

The odds ratio estimates specify the following `model_2` predictions...

- Subjects with a stent have 1.78 times the odds of being in the abcix = 1 group (treated with abcix) than subjects without a stent who have the same ejection fraction and same level of the `ves_fix` variable. The 95% CI for that odds ratio is (1.33, 2.40).
- Suppose Harry and Larry have the same status in terms of `stent` and `ves_fix` but Harry's ejection fraction is one percentage point larger than Larry's. Harry's predicted odds of `abcix` treatment will be

97.4% of Larry's, with a 95% CI of (0.959, 0.989).

- Subjects with a Middle level of `ves_fix` have 2.27 times the odds (with 95% CI 1.59, 3.28) of receiving abcix treatment as compared to subjects in the Low `ves_fix` group, assuming that they have the same `stent` status and ejection fraction.
- Subjects with High `ves_fix` have 5.91 times the odds of receiving `abcix` (with 95% CI 2.53, 17.30) as compared to Low `ves_fix` subjects with the same `stent` and `ejecfrac` values.

Make a prediction from `model_2`

Suppose we have a subject named Harry with `ejecfrac` = 60%, with a `stent` and a High `ves_fix`. What is that subject's predicted probability of being treated with `abcix`?

```
harry <- tibble(ejecfrac = 60, stent = 1, ves_fix = "High")

predict(model_2, newdata = harry, type = "response")
```

```
        1
0.9167082
```

Note that `model_2` is a `glm` model, and so we use `type = "response"` to get a predicted probability. As we'll see later, if our logistic model had been fit using `lrm`, we'd have to select a different type of prediction in order to get a predicted probability.

Confusion Matrix for `model_2`

What are the predicted probabilities of `abcix = 1` that we get from `model_2`?

```
mod2_aug <- augment(model_2, type.predict = "response")

summary(mod2_aug$.fitted)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4139  0.6505  0.7015  0.7082  0.8084  0.9669
```

Let's build a confusion matrix with decision rule "we predict that the subject received `abcix` if the predicted probability that (`abcix` = 1) is greater than or equal to 0.5"

```
mod2_aug %$%
    confusionMatrix(
        data = factor(.fitted >= 0.5),
        reference = factor(abcix == 1),
        positive = "TRUE"
    )
```

```
Confusion Matrix and Statistics

          Reference
Prediction FALSE TRUE
     FALSE     8    3
     TRUE    275  684

               Accuracy : 0.7134
                 95% CI : (0.6838, 0.7417)
    No Information Rate : 0.7082
    P-Value [Acc > NIR] : 0.377

                  Kappa : 0.0333
```

```
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.99563
            Specificity : 0.02827
         Pos Pred Value : 0.71324
         Neg Pred Value : 0.72727
             Prevalence : 0.70825
         Detection Rate : 0.70515
   Detection Prevalence : 0.98866
      Balanced Accuracy : 0.51195

       'Positive' Class : TRUE
```

With a decision rule setting our cutoff for a positive prediction at 0.5, we have over 99% sensitivity but only 3% specificity. The sum is $0.996 + 0.028 = 1.024$. Can we do meaningfully better with a different rule?

Comparing Decision Rules

Let's try a decision rule with a cutoff of 0.7 instead.

```
mod2_aug %$%
    confusionMatrix(
        data = factor(.fitted >= 0.7),
        reference = factor(abcix == 1),
        positive = "TRUE"
    )
```

```
Confusion Matrix and Statistics

          Reference
Prediction FALSE TRUE
     FALSE   178  283
     TRUE    105  404

               Accuracy : 0.6
                 95% CI : (0.5684, 0.631)
    No Information Rate : 0.7082
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1832

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.5881
            Specificity : 0.6290
         Pos Pred Value : 0.7937
         Neg Pred Value : 0.3861
             Prevalence : 0.7082
         Detection Rate : 0.4165
   Detection Prevalence : 0.5247
      Balanced Accuracy : 0.6085

       'Positive' Class : TRUE
```

I've run a few other options, too, and the results are summarized below.

Cutpoint	Sensitivity	Specificity	Sens + Spec
0.5	0.996	0.028	1.024
0.6	0.863	0.286	1.149
0.7	0.588	0.629	1.217
0.8	0.325	0.862	1.187
0.9	0.060	0.982	1.042

So it seems like 0.7 might be a reasonable decision rule here if we care equally about sensitivity and specificity.

Plotting the ROC curve for `model_2`: A Simple Strategy using the `pROC` package

```
## requires pROC package
out_m2 <- lind_new$abcix # meant to be a 1-0 numeric variable
prob_m2 <- predict(model_2, type="response") # predictions on 0-1 scale

roc_m2 <- roc(out_m2 ~ prob_m2)
```

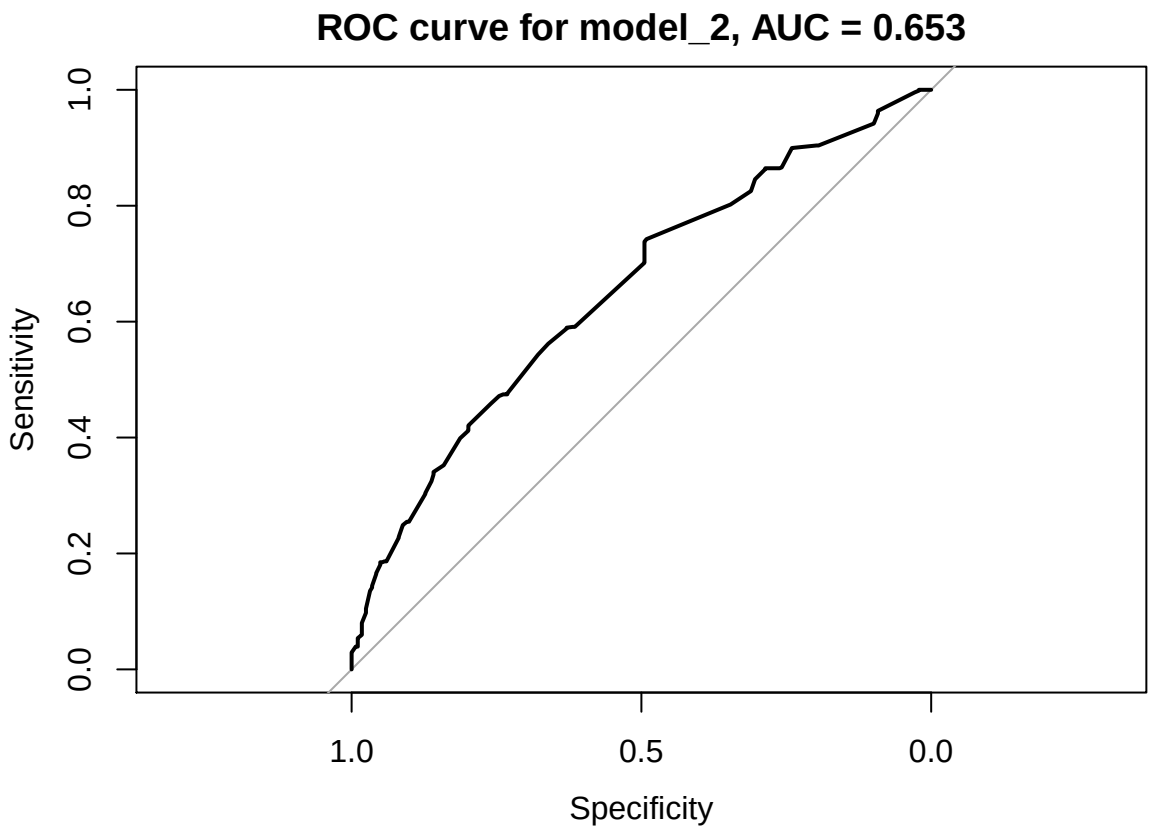
```
Setting levels: control = 0, case = 1
```

```
Setting direction: controls < cases
```

```
auc(roc_m2) # tell me the C statistic
```

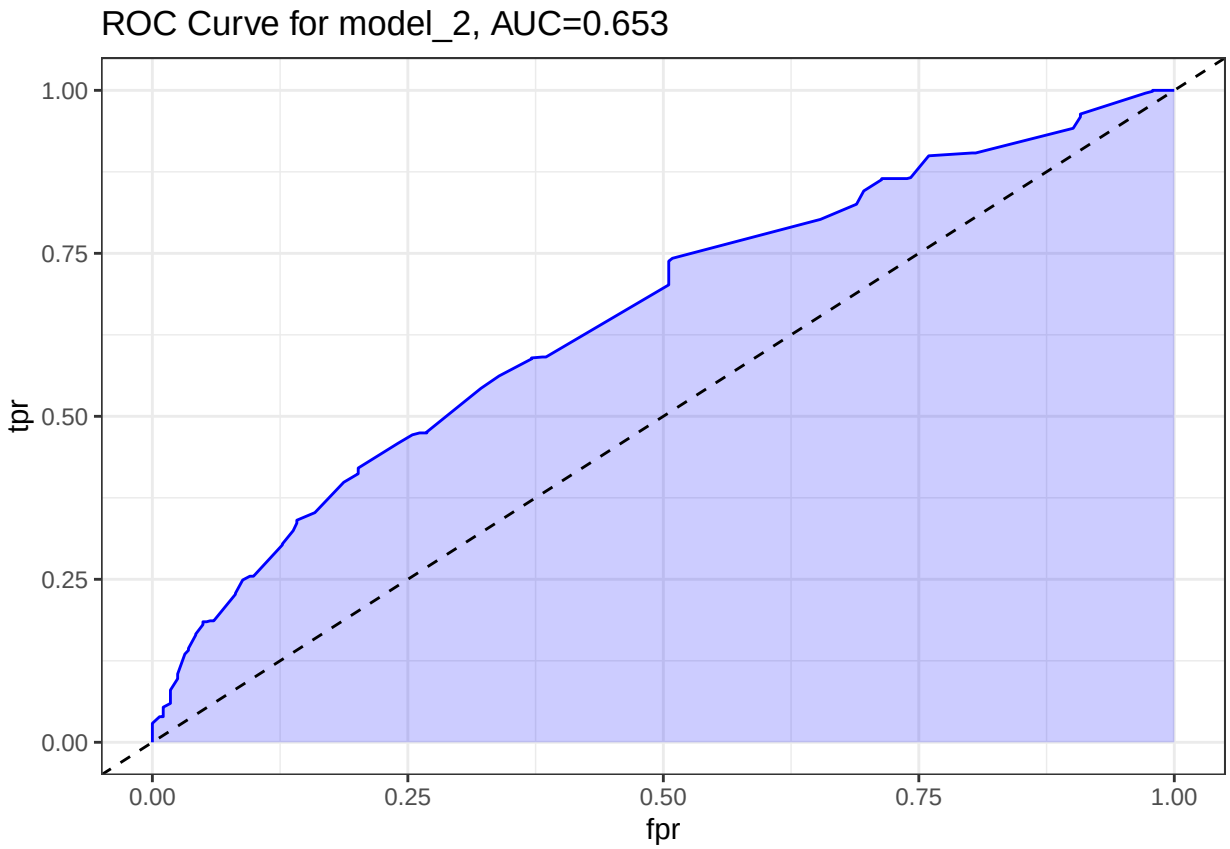
```
Area under the curve: 0.6531
```

```
plot(roc_m2,
     main = paste0("ROC curve for model_2, AUC = ",
                   round(auc(roc_m2),3)))
```



Plotting the ROC curve for `model_2`: Using the `ROCR` package

```
## requires ROCR package
prob <- predict(model_2, type="response")
pred <- prediction(prob, lind_new$abcix)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
auc <- performance(pred, measure="auc")

auc <- round(auc@y.values[[1]],3)
roc.data <- data.frame(fpr=unlist(perf@x.values),
                       tpr=unlist(perf@y.values),
                       model="GLM")

ggplot(roc.data, aes(x=fpr, ymin=0, ymax=tpr)) +
    geom_ribbon(alpha=0.2, fill = "blue") +
    geom_line(aes(y=tpr), col = "blue") +
    geom_abline(intercept = 0, slope = 1, lty = "dashed") +
    labs(title = paste0("ROC Curve for model_2, AUC=", auc)) +
    theme_bw()
```



`model_3`: An augmented model

We'll add a spline with four knots in `ejecfrac` and an interaction between the main effect of `ejecfrac` and `stent`. When fitting this model, I will always do most of the work using `lrm`.

```
dd <- datadist(lind_new)
options(datadist = "dd")

model_3 <- lrm(abcix ~ rcs(ejecfrac, 4) + stent +
                  ejecfrac %ia% stent + ves_fix,
               data = lind_new, x = TRUE, y = TRUE)

model_3
```

```
Logistic Regression Model

 lrm(formula = abcix ~ rcs(ejecfrac, 4) + stent + ejecfrac %ia%
     stent + ves_fix, data = lind_new, x = TRUE, y = TRUE)

                       Model Likelihood     Discrimination    Rank Discrim.
                          Ratio Test           Indexes           Indexes
 Obs            970    LR chi2      67.75    R2        0.096    C         0.656
  0             283    d.f.             7    g         0.721    Dxy       0.312
  1             687    Pr(> chi2) <0.0001    gr        2.056    gamma     0.325
 max |deriv| 1e-09                          gp        0.133    tau-a     0.129
                                            Brier     0.193
```

```
                 Coef    S.E.   Wald Z Pr(>|Z|)
Intercept         0.3014 0.8670  0.35  0.7281
ejecfrac          0.0034 0.0214  0.16  0.8750
ejecfrac'        -0.0112 0.0278 -0.40  0.6869
ejecfrac''       -0.1226 0.4102 -0.30  0.7650
stent             1.4846 0.8223  1.81  0.0710
ejecfrac * stent -0.0173 0.0156 -1.11  0.2665
ves_fix=Mid       0.8110 0.1837  4.41  <0.0001
ves_fix=High      1.7934 0.4807  3.73  0.0002
```

Summarizing the Effect Sizes

```
summary(model_3)
```

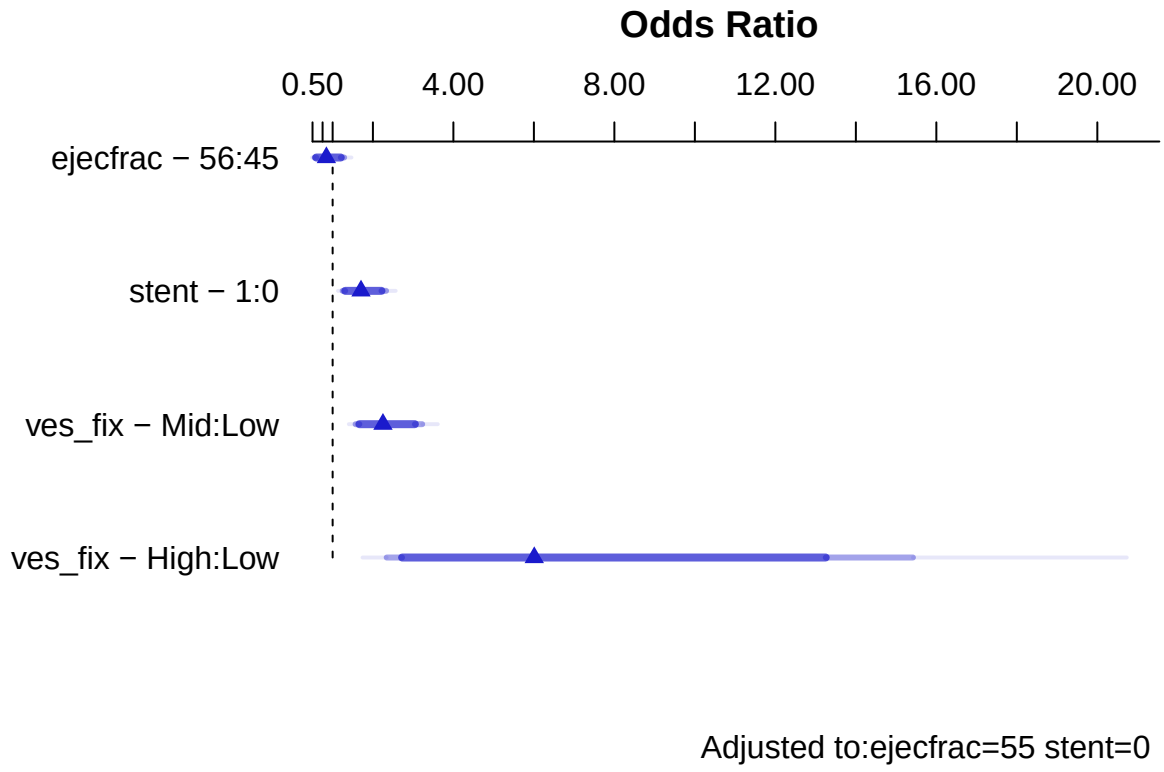
```
             Effects              Response : abcix

 Factor             Low High Diff. Effect   S.E.    Lower 0.95 Upper 0.95
 ejecfrac           45  56   11    -0.16899 0.21475 -0.58990    0.25191
  Odds Ratio        45  56   11     0.84451      NA  0.55438    1.28650
 stent               0   1    1     0.53330 0.15889  0.22187    0.84472
  Odds Ratio         0   1    1     1.70450      NA  1.24840    2.32730
 ves_fix - Mid:Low   1   2   NA     0.81101 0.18372  0.45092    1.17110
  Odds Ratio         1   2   NA     2.25020      NA  1.56980    3.22560
 ves_fix - High:Low  1   3   NA     1.79340 0.48073  0.85114    2.73560
  Odds Ratio         1   3   NA     6.00960      NA  2.34230   15.41900

Adjusted to: ejecfrac=55 stent=0
```

Interpreting those results, we see that in `model_3`...

- If we have two subjects, each with `stent` = 0 and the same status for `ves_fix`, but Harry has an ejection fraction of 56, while Larry's is 45, then Harry is predicted to have 0.845 times the odds of being treated with `abcix` that Larry does. The 95% CI for the odds ratio is (0.554, 1.287).
  - Note that because of the interaction term between `stent` and `ejecfrac`, we have to look at the bottom to see which value of `stent` we're adjusting the results to in order to interpret the `ejecfrac` odds ratio.
- If we have two subjects each with `ejecfrac` = 55, but Harry has a stent and Barry does not, then Harry is predicted to have 1.70 times the odds of being treated with `abcix` that Barry does. The 95% CI for the odds ratio is (1.25, 2.33).
  - Again, because of the interaction between `stent` and `ejecfrac`, we cannot interpret the `stent` effect until we specify the level of `ejecfrac`, which we read off from the "Adjusted to:" section of the output.
- If we have two subjects with the same values of `ejecfrac` and the same `stent` status, then having Middle `ves_fix` is associated with 2.25 times the odds of `abcix` treatment as compared to Low `ves_fix`, with 95% CI (1.57, 3.23) for the odds ratio.
  - Since `ves_fix` is not included in a product term with the other variables, we can simply assume those other variables are the same, and do not, for instance, have to assume they are exactly equal to the values in the Adjusted to section.
- Finally, if we have two subjects with the same values of `ejecfrac` and the same `stent` status, then having High `ves_fix` is associated with 6.01 times the odds of `abcix` treatment as compared to Low `ves_fix`, with 95% CI (2.34, 15.42) for the odds ratio.

Here is the plot of those effect size results...

```
plot(summary(model_3))
```



ANOVA for `model_3`

Here's the ANOVA table for `model_3`. It doesn't appear that the non-linear + interaction terms added statistically detectable predictive value.

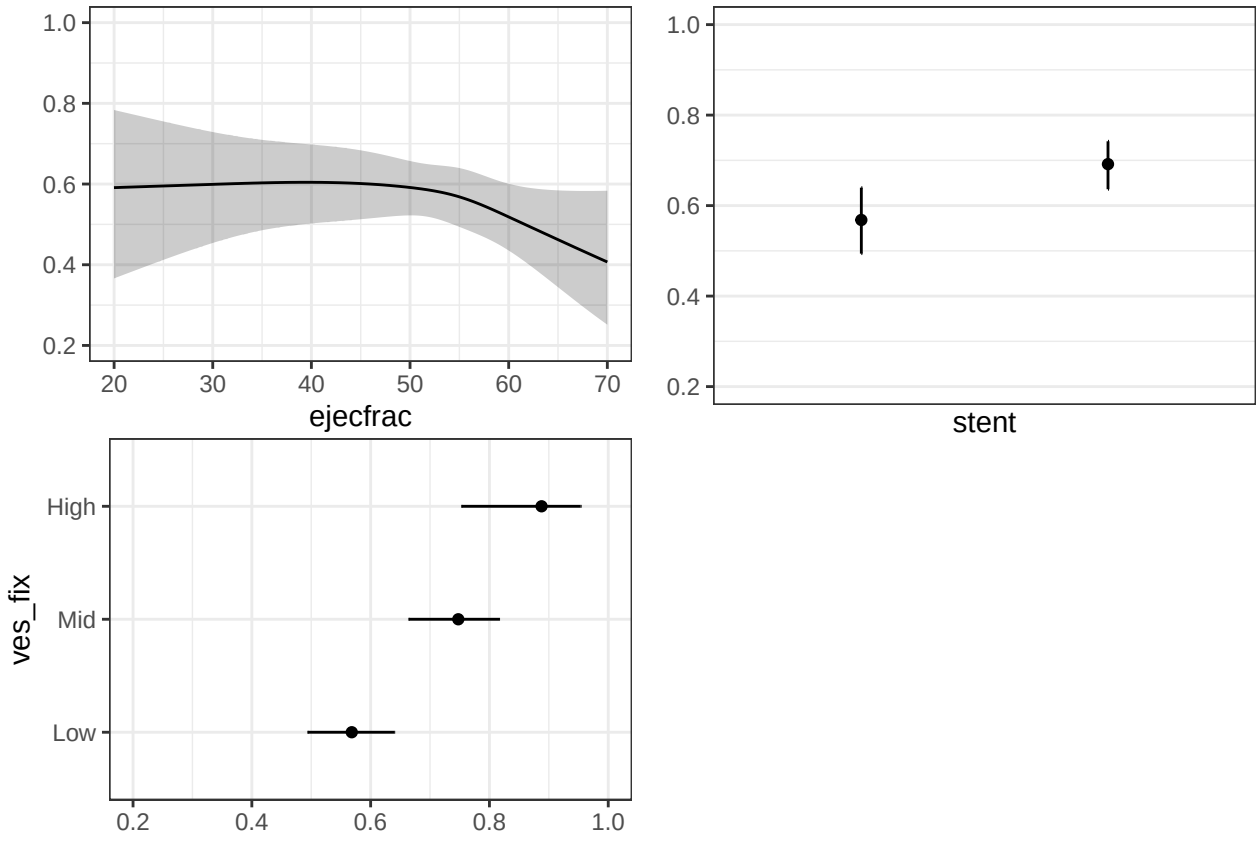
```
anova(model_3)
```

```
                Wald Statistics          Response: abcix

 Factor                                                Chi-Square d.f. P
 ejecfrac  (Factor+Higher Order Factors)               15.37      4    0.0040
  All Interactions                                      1.23      1    0.2665
  Nonlinear                                             2.81      2    0.2449
 stent  (Factor+Higher Order Factors)                  16.21      2    0.0003
  All Interactions                                      1.23      1    0.2665
 ejecfrac * stent  (Factor+Higher Order Factors)        1.23      1    0.2665
 ves_fix                                               31.04      2    <.0001
 TOTAL NONLINEAR + INTERACTION                          4.44      3    0.2181
 TOTAL                                                 57.63      7    <.0001
```

Plotting the Predicted Values and Confidence Limits at Each Coefficient

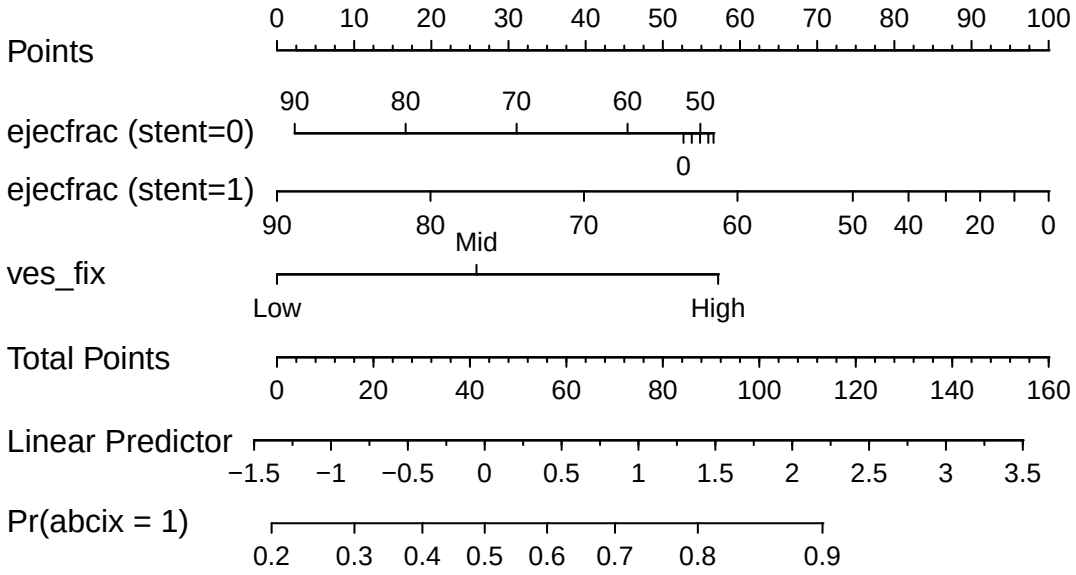
The predicted values and confidence limits at each level of `ejecfrac`, `stent` and `ves_fix` implied by `model_3` are of some interest. I like to plot these in terms of the probabilities of experiencing our outcome (treatment with `abcix`) so I use the `fun = plogis` code below to help with that.

```
ggplot(Predict(model_3, fun = plogis))
```



The Nomogram for `model_3`

```
plot(nomogram(model_3, fun = plogis, funlabel = "Pr(abcix = 1)"))
```



Validation of `model_3` summary statistics

```
set.seed(432)

validate(model_3, B = 100)
```

```
          index.orig training   test optimism index.corrected   n
Dxy           0.3123   0.3282 0.3060   0.0221          0.2901 100
R2            0.0962   0.1068 0.0883   0.0184          0.0778 100
Intercept     0.0000   0.0000 0.0750  -0.0750          0.0750 100
Slope         1.0000   1.0000 0.8979   0.1021          0.8979 100
Emax          0.0000   0.0000 0.0370   0.0370          0.0370 100
D             0.0688   0.0768 0.0629   0.0139          0.0549 100
U            -0.0021  -0.0021 0.0011  -0.0032          0.0011 100
Q             0.0709   0.0789 0.0618   0.0171          0.0538 100
B             0.1929   0.1907 0.1943  -0.0037          0.1965 100
g             0.7207   0.7715 0.6806   0.0910          0.6298 100
gp            0.1331   0.1377 0.1253   0.0123          0.1208 100
```

- Our validated C statistic is $0.5 + (0.2901/2) = 0.645$
- Our validated Nagelkerke R-square is 0.078.

Make a prediction from `model_3`

Suppose we have a subject named Harry with `ejecfrac` = 60%, with a `stent` and a High `ves_fix`. What is that subject's predicted probability of being treated with `abcix`?

Since we have a spline here and an interaction term using that spline, we will need to use the `ols` model to fit our prediction, and that requires the approach below.

```
harry <- tibble(ejecfrac = 60, stent = 1, ves_fix = "High")

predict(model_3, newdata = harry, type = "fitted")
```

```
       1
0.910042
```

Run `model_3` with `glm` to get some other pieces

```
model_3glm <- lind_new %$%
    glm(abcix ~ rcs(ejecfrac, 4) + stent + ejecfrac %ia% stent + ves_fix,
        family = binomial)
```

Confusion Matrix for `model_3glm`

What are the predicted probabilities of `abcix = 1` that we get from `model_3glm`?

```
model_3glm_aug <- augment(model_3glm, type.predict = "response")

summary(model_3glm_aug$.fitted)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2057  0.6043  0.7199  0.7082  0.7911  0.9594
```

Let's build a confusion matrix with decision rule "we predict that the subject received `abcix` if the predicted probability that (`abcix` = 1) is greater than or equal to 0.5"

```
model_3glm_aug %$%
    confusionMatrix(
        data = factor(.fitted >= 0.9),
        reference = factor(abcix == 1),
        positive = "TRUE"
    )
```

```
Confusion Matrix and Statistics

          Reference
Prediction FALSE TRUE
     FALSE   279  650
     TRUE      4   37

               Accuracy : 0.3258
                 95% CI : (0.2963, 0.3563)
    No Information Rate : 0.7082
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0238

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.05386
            Specificity : 0.98587
         Pos Pred Value : 0.90244
```

```
         Neg Pred Value : 0.30032
             Prevalence : 0.70825
         Detection Rate : 0.03814
   Detection Prevalence : 0.04227
      Balanced Accuracy : 0.51986

       'Positive' Class : TRUE
```

Comparing Decision Rules

Results from trying varying cutpoints in `model_3glm` prediction follow:

Cutpoint	Sensitivity	Specificity	Sens + Spec
0.5	0.993	0.056	1.049
0.6	0.836	0.311	1.147
0.7	0.583	0.636	1.219
0.8	0.250	0.912	1.162
0.9	0.054	0.986	1.040

So it again seems like 0.7 might be a reasonable decision rule here if we care equally about sensitivity and specificity.

Plotting the ROC curve for `model_3glm`: A Simple Strategy using the `pROC` package

```
## requires pROC package
out_m3glm <- lind_new$abcix # meant to be a 1-0 numeric variable
prob_m3glm <- predict(model_3glm, type="response") # predictions on 0-1 scale

roc_m3glm <- roc(out_m3glm ~ prob_m3glm)
```

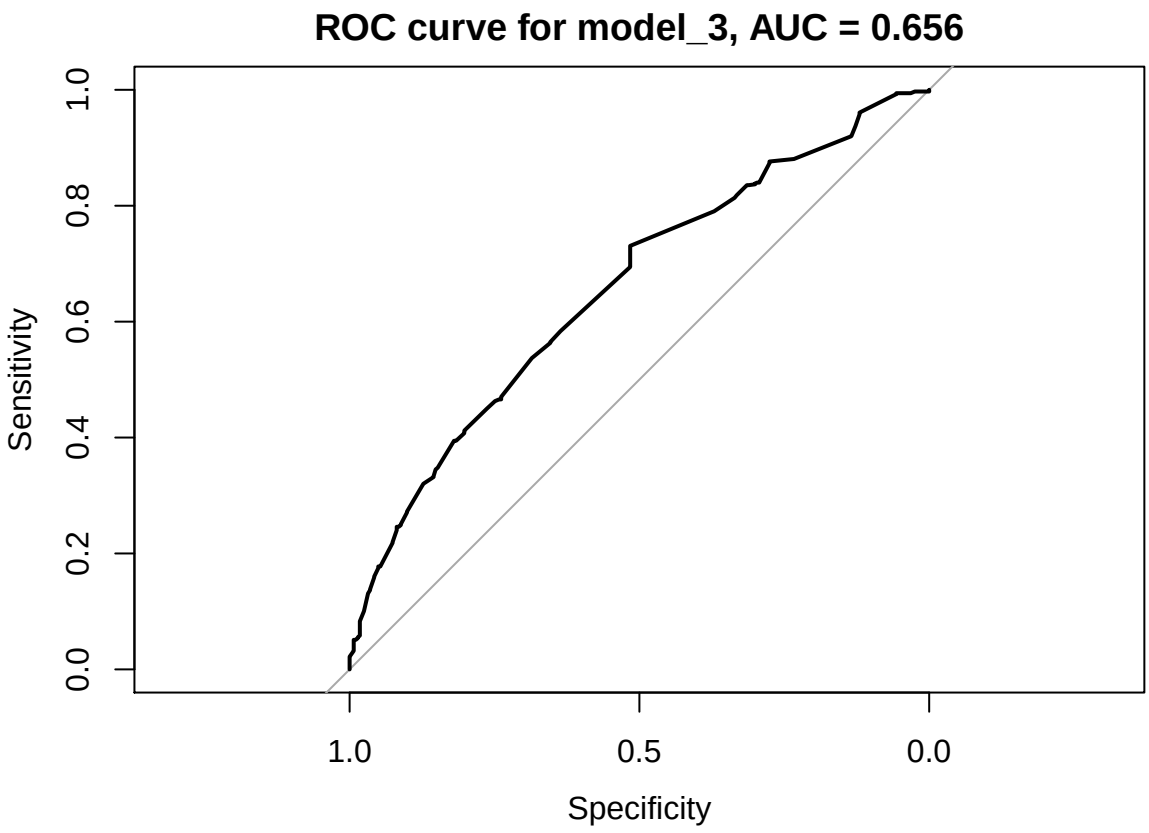
```
Setting levels: control = 0, case = 1
```

```
Setting direction: controls < cases
```

```
auc(roc_m3glm) # tell me the C statistic
```

```
Area under the curve: 0.6561
```

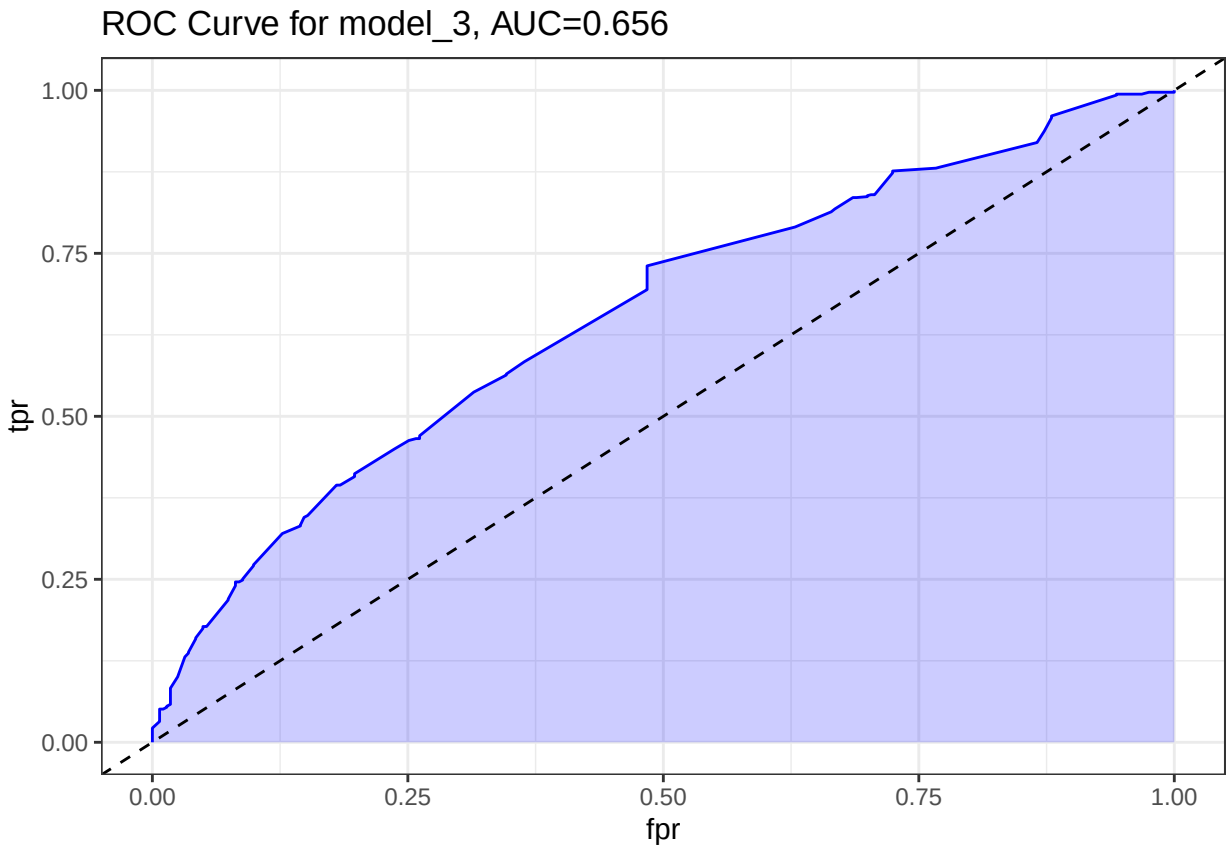
```
plot(roc_m3glm,
     main = paste0("ROC curve for model_3, AUC = ",
                   round(auc(roc_m3glm),3)))
```



Note that this matches the value produced by the `lrm` package for model 3. This plots the original ROC curve, not the results after validation.

Plotting the ROC curve for `model_3glm`: Using the `ROCR` package

```
## requires ROCR package
prob <- predict(model_3glm, type="response")
pred <- prediction(prob, lind_new$abcix)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
auc <- performance(pred, measure="auc")

auc <- round(auc@y.values[[1]],3)
roc.data <- data.frame(fpr=unlist(perf@x.values),
                       tpr=unlist(perf@y.values),
                       model="GLM")

ggplot(roc.data, aes(x=fpr, ymin=0, ymax=tpr)) +
    geom_ribbon(alpha=0.2, fill = "blue") +
    geom_line(aes(y=tpr), col = "blue") +
    geom_abline(intercept = 0, slope = 1, lty = "dashed") +
    labs(title = paste0("ROC Curve for model_3, AUC=", auc)) +
    theme_bw()
```


ROC Curve for model_3, AUC=0.656
tpr
fpr
0.00
0.25
0.50
0.75
1.00