# Homework 10

Advanced Methods for Data Analysis (36-402)

Due Friday April 22, 2022 at 3:00pm EDT

**Solutions** – not to be posted online or shared, even after the end of the semester.

You should **always show all your work** and submit both a writeup and *R* code.

- Assignments must be submitted through Gradescope as a PDF. Follow the instructions here: `https://www.cmu.edu/teaching/gradescope/`
- Gradescope will ask you to mark which parts of your submission correspond to each homework problem. This is mandatory; if you do not, grading will be slowed down, and your assignment will be penalized.
- Make sure your work is legible in Gradescope. You may not receive credit for work the TAs cannot read. **Note:** If you submit a PDF with pages much larger than $8.5 \times 11$", they will be blurry and unreadable in Gradescope.
- For questions involving R code, we strongly recommend using R Markdown. The relevant code should be included with each question, rather than in an appendix. A template Rmd file is provided on Canvas.

1. **Diabetes among Native Americans.**

   The data set we will use for this problem contains information on 768 female Pima people from Arizona. It is posted in the file `pima.csv`. There are nine variables, including a test for the presence of diabetes:

   **pregnant:** Number of times pregnant

   **glucose:** Plasma glucose concentration at 2 hours in an oral glucose tolerance test

   **diastolic:** Diastolic blood pressure (mm Hg)

   **triceps:** Triceps skin fold thickness (mm)

   **insulin:** 2-hour serum insulin (mu U/ml)

   **bmi:** Body mass index (weight in kg/(height in metres squared))

   **diabetes:** Diabetes pedigree function (a measure of family history of diabetes)

   **age:** Age (years)

   **test:** test whether the patient shows signs of diabetes (coded 0 if negative, 1 if positive)

   We will model the response `test` using logistic regression on all or some of the other measured variables.

   The `glm` function in R will fit a logistic regression. For example, suppose that we want to use only `bmi` and `insulin` as predictors:

   ```
   glm(test ~ bmi + insulin, family = binomial, data = pima)
   ```

   will return a logistic regression fit with maximum likelihood estimates of the parameters.

(a) Compute summaries and create plots of the data to look for patterns. Comment on any patterns and relationships you see. In particular, you might notice that almost every variable has a minimum value of 0. For some variables (such as `test` or `pregnant`) a value of 0 is easy to understand. What could a value of 0 for `bmi` or `diastolic` or `triceps` mean?

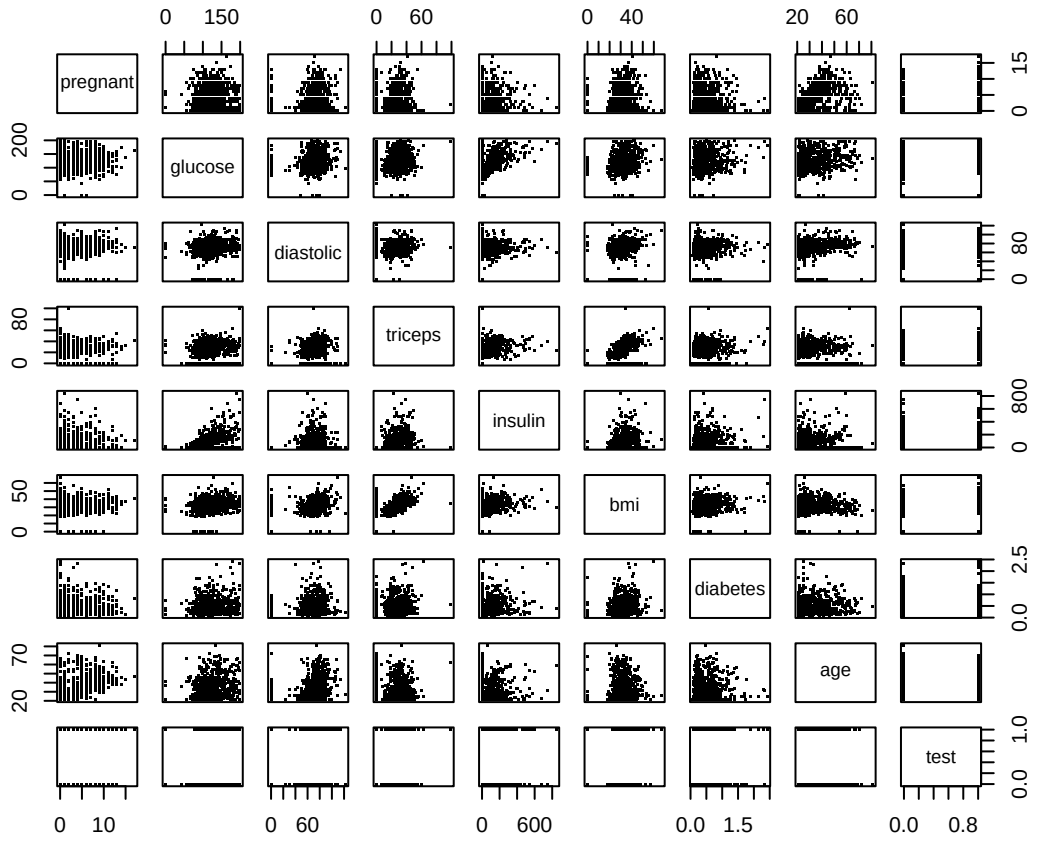
Replace all 0 values of the following variables with NA: `glucose`, `diastolic`, `triceps`, `insulin`, and `bmi`. Create a new `data.frame` containing only those observations that have no NA values. Use this reduced `data.frame` for the rest of the questions. There should be 392 observations in the resulting `data.frame`.

**Solution:** We start with a data summary and plots. The instances of 0 for variables like `triceps` and `bmi` are impossible. It seems more likely that 0 was used a missing data indicator.

```
pima <- read.csv("pima.csv", header = TRUE)
summary(pima)
##     pregnant         glucose        diastolic         triceps
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     insulin           bmi           diabetes           age
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00
##  Median : 30.5   Median :32.00   Median :0.3725   Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00
##       test
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

A pairs plot shows the zeros and the relationships:

```
pairs(pima, pch = ".")
```



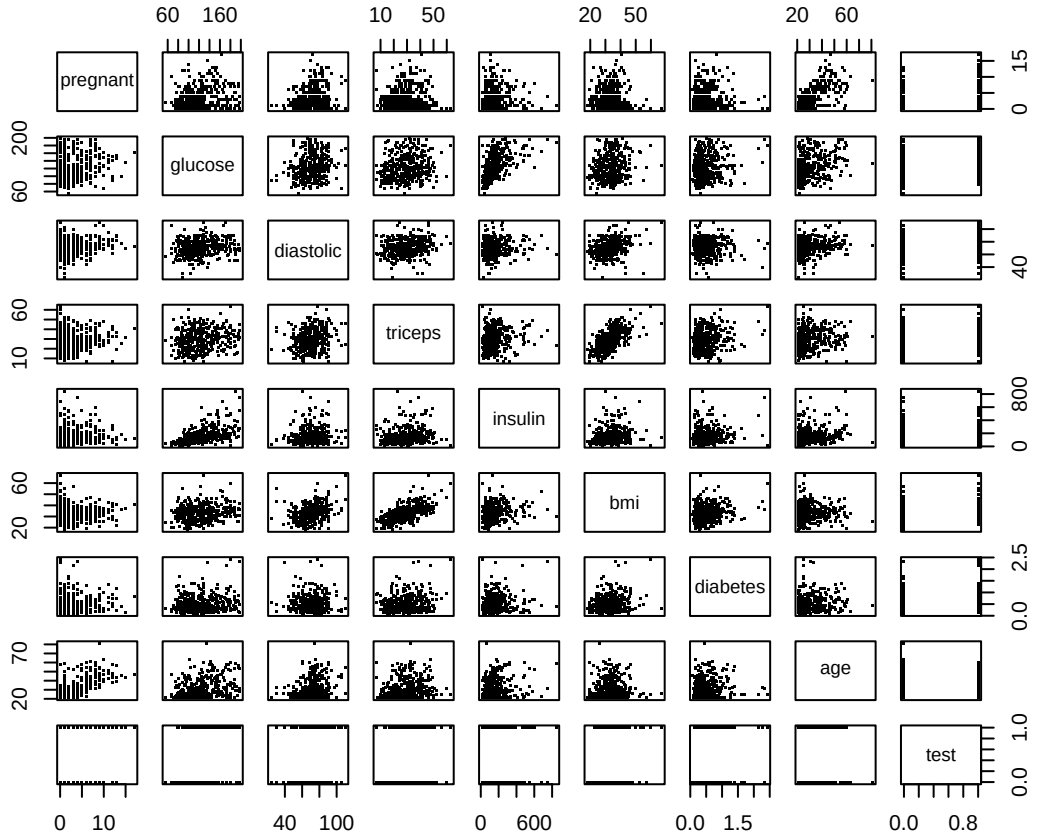

We replace the zeros with `NA` and then take only complete cases for the rest of the assignment.

```
pima$glucose[pima$glucose == 0] <- NA
pima$diastolic[pima$diastolic == 0] <- NA
pima$triceps[pima$triceps == 0] <- NA
pima$insulin[pima$insulin == 0] <- NA
pima$bmi[pima$bmi == 0] <- NA

Pima <- pima[complete.cases(pima), ]
nrow(Pima)
## [1] 392
```

It is difficult to see a relationship between the binary variable `test` and any of the continuous variables in scatter plots, even after removing the missing data.

```
pairs(Pima, pch = ".")
```




We can see that several of the continuous-valued predictors are correlated with each other. There may be some predictors whose effects on `test` are being masked by other predictors.

(b) Start by using all eight of the predictors in a logistic regression model. Call this Model 1. Report a summary of the fit. Does it appear that all of the variables are contributing to the fit?

**Solution:**

```
Model1 <- glm(test ~ ., family = binomial, data = Pima)
summary(Model1)
##
## Call:
## glm(formula = test ~ ., family = binomial, data = Pima)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7823  -0.6603  -0.3642   0.6409   2.5612
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.004e+01  1.218e+00  -8.246  < 2e-16 ***
## pregnant     8.216e-02  5.543e-02   1.482  0.13825
## glucose      3.827e-02  5.768e-03   6.635 3.24e-11 ***
## diastolic   -1.420e-03  1.183e-02  -0.120  0.90446
## triceps      1.122e-02  1.708e-02   0.657  0.51128
## insulin     -8.253e-04  1.306e-03  -0.632  0.52757
## bmi          7.054e-02  2.734e-02   2.580  0.00989 **
## diabetes     1.141e+00  4.274e-01   2.669  0.00760 **
## age          3.395e-02  1.838e-02   1.847  0.06474 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 498.10  on 391  degrees of freedom
## Residual deviance: 344.02  on 383  degrees of freedom
## AIC: 362.02
##
## Number of Fisher Scoring iterations: 5
```

There are several variables with statistically insignificant coefficients. Only `glucose`, and possibly `bmi` and `diabetes` look like they are significantly impacting the fit. Of course, many variables are correlated with others such as `pregnant` and `age`. Even though neither of these has a significant coefficient, they could each be masking the other's effect.

(c) Fit a model in which no predictors appear, that is, every observation has the same distribution for `test` (the `formula` for `glm` is `test ~ 1`). Call this Model 2. Test whether Model 1 is a significant improvement on Model 2.

**Solution:** The model with no predictors simply assigns the fraction of $Y = 1$ cases as the fitted value for each observation.

```
Model2 <- glm(test ~ 1, family = binomial, data = Pima)

unique(fitted.values(Model2))
## [1] 0.3316327
mean(Pima$test)
## [1] 0.3316327
```

In part (i) we will be asked to verify that the only fitted value of Model 2 is the fraction of $Y = 1$ cases. The calculations to verify that claim are printed right after the model fit above. The common value is 0.3316.

We then conduct a deviance test against Model 1:

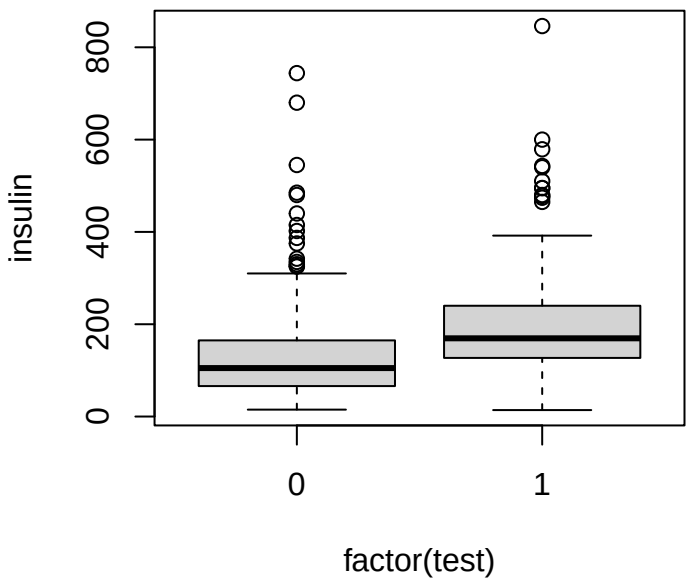
```
anova(Model2, Model1, test = "Chisq")
## Analysis of Deviance Table
##
## Model 1: test ~ 1
## Model 2: test ~ pregnant + glucose + diastolic + triceps + insulin + bmi +
##     diabetes + age
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       391     498.10
## 2       383     344.02  8   154.08 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The deviance test suggests that Model 1 makes a significant improvement over Model 2.

(d) Do women with signs of diabetes have higher 2-hour serum insulin values? Is the **insulin** coefficient significant and positive in Model 1? Explain why these answers are not contradictory.

**Solution:**

```
plot(insulin ~ factor(test), data = Pima)
```




The boxplot shows that the `insulin` values are generally higher amongst the women who show signs of diabetes. However, the coefficient of `insulin` in the Model 1 fit is negative ($-8.253 \times 10^{-4}$) and not significant. The reason that these facts are not contradictory is that `insulin` seems to be positively related to `glucose`, `bmi`, and `diabetes`, all of which have positive and (at least somewhat) significant coefficients. The negative coefficient on `insulin` might just be compensating for some of the combined effects of the other variables.

(e) Fit a third model, Model 3, by running backward elimination on Model 1. To fit Model 3, run the `step` function with the argument `direction="backward"` as in Demo 10.2; the `step` function will choose to stop eliminating when AIC is minimized.

Compare Model 3 to Model 1 using a deviance test. Which model do the tests in this part and part (c) suggest describes the data best?

**NOTE:** For this part and all other parts of the assignment that involve the `step` command, **do not** include the spew of backward elimination output in your submission. If some part of that output is useful to your analysis, extract it separately. The argument `trace=0` will suppress printing of the intermediate output.

**Solution:** We will use the `step` command to fit another model via backward elimination from Model 1.

```
Model3 <- step(Model1, direction = "backward", trace = 0)
summary(Model3)
##
## Call:
## glm(formula = test ~ pregnant + glucose + bmi + diabetes + age,
##     family = binomial, data = Pima)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8827  -0.6535  -0.3694   0.6521   2.5814
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.992080   1.086866  -9.193  < 2e-16 ***
## pregnant     0.083953   0.055031   1.526 0.127117
## glucose      0.036458   0.004978   7.324 2.41e-13 ***
## bmi          0.078139   0.020605   3.792 0.000149 ***
## diabetes     1.150913   0.424242   2.713 0.006670 **
## age          0.034360   0.017810   1.929 0.053692 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 498.10  on 391  degrees of freedom
## Residual deviance: 344.89  on 386  degrees of freedom
## AIC: 356.89
##
## Number of Fisher Scoring iterations: 5
```

Now we test against Model 1:

```
dev.test <- anova(Model3, Model1, test = "Chisq")
dev.test
## Analysis of Deviance Table
##
## Model 1: test ~ pregnant + glucose + bmi + diabetes + age
## Model 2: test ~ pregnant + glucose + diastolic + triceps + insulin + bmi +
##     diabetes + age
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       386     344.89
## 2       383     344.02  3   0.8639   0.8341
```

It appears that Model 1 does not improve significantly on Model 3. For completeness, we also compare Model 2 to Model 3.

```
anova(Model2, Model3, test = "Chisq")
## Analysis of Deviance Table
##
## Model 1: test ~ 1
## Model 2: test ~ pregnant + glucose + bmi + diabetes + age
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       391     498.10
## 2       386     344.89  5   153.21 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Not surprisingly, Model 3 improves significantly on Model 2.

(f) Model 3 in part (e) was chosen by backward elimination. The deviance test does not take into account that one of the models being compared was chosen by backward elimination, which could have selected different variables with a different sample of data. As a result, the nominal asymptotic $\chi^2$ distribution of the deviance test statistic under the null hypothesis might be incorrect.

Perform a bootstrap analysis to determine the appropriate distribution of the deviance test statistic for comparing Models 1 and 3. Use a parametric bootstrap in which each bootstrap sample $b$ is drawn as follows: For each predictor $x_i$, draw $Y_{b,i}$ from a Bernoulli distribution, with probability of success equal to what Model 3 predicts. Then refit both models, **including** the backward selection step.

Repeat the bootstrap sampling $B = 1000$ times. Plot the distribution of deviance test statistics, mark the value you obtained in part (e) on the plot, and report the bootstrap $p$ value.

**Note:** Make sure to use the argument `trace=0` in `step` to avoid seeing the output of 1000 backward eliminations. Also, the bootstrap may take a few minutes.

**Solution:** We will use the `rbinom` function with `size` argument equal to 1 to simulate $n = 392$ Bernoulli random variables with different success probablities as determined by Model 3.

```
probs <- fitted.values(Model3)
thedata <- Pima
teststats <- numeric(1000)
n <- nrow(Pima)

for (b in 1:1000) {
    thedata$test <- rbinom(n, 1, probs)
    m1 <- glm(test ~ ., family = binomial, data = thedata)
    m3 <- step(m1, direction = "backward", trace = 0)
    teststats[b] <- anova(m3, m1, test="Chisq")$Deviance[2]
}
```

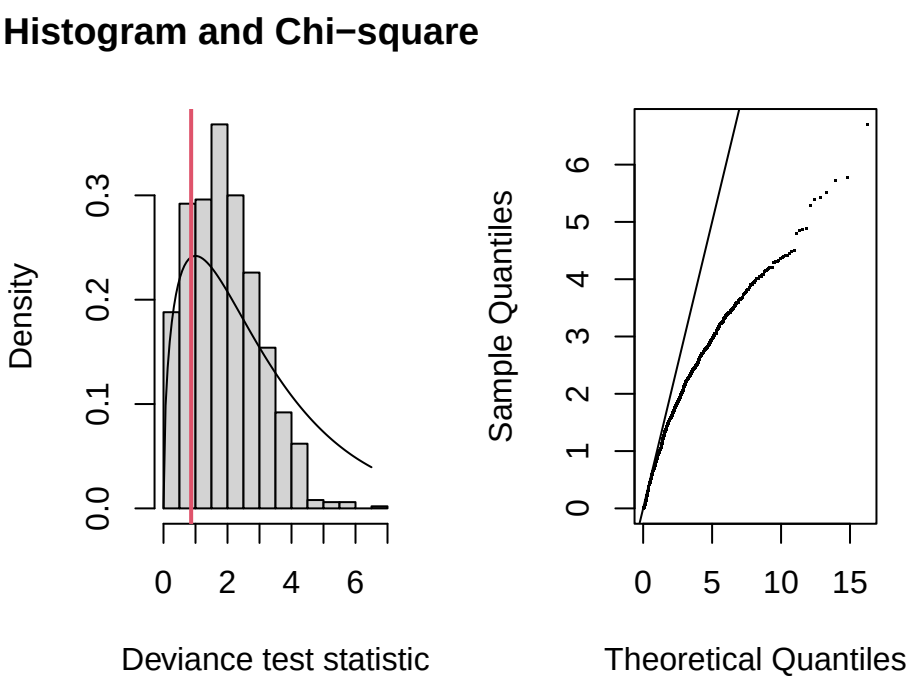
The bootstrap $p$ value is

```
mean(teststats >= dev.test$Deviance[2])
## [1] 0.802
```

Now let's compare the distribution of deviance test statistics against what we would expect from a $\chi^2$ distribution. The vertical line indicates the deviance test statistic observed in part (e).

```
par(mfrow=c(1,2))
xs <- seq(0, 6.5, length.out = 101)
hist(teststats, freq=FALSE, xlab="Deviance test statistic",
     main="Histogram and Chi-square pdf")
lines(xs, dchisq(xs, 3))
abline(v = dev.test$Deviance[2], lwd = 2, col = 2)

plot(qchisq(c(1:1000)/1001, 3), sort(teststats), pch=".",
     xlab="Theoretical Quantiles", ylab="Sample Quantiles")
abline(0, 1)
```




The nominal asymptotic null distribution of the deviance test statistic would be $\chi^2$ with 3 degrees of freedom. The histogram of the bootstrap test statistics is similar to the $\chi^2$ density, but the bootstrap distribution has a thinner upper tail, with more of its mass between 1 and 3. The thinner upper tail is even more apparent in the quantile plot, where all of the theoretical quantiles are larger than the sample quantiles. Despite the clear differences between the distributions, the $p$-values that correspond to the observed deviance test statistic are rather close.

(g) Consider a Pima woman who has been pregnant 3 times, has a glucose concentration of 107, diastolic pressure of 70, 29.2mm of tricep thickness, 2-hour serum insulin level of 160, body mass index of 32.4, diabetes pedigree function of 0.6, and is 34 years old. Predict her `test` result by computing the probability of positive `test` result based on Model 3. Also give a 90% confidence interval for the Model 3 probability of a positive test result.

**Solution:** There are two ways to compute a confidence interval for the requested probability. The naive way is to do all of the calculations in the "response" (probability) scale. A somewhat better way is to do the calculations in the "link" (log-odds) scale and then transform back to

the response scale. We will do both here.

The new subject is

```
newx <- data.frame(pregnant=3, glucose=107, diastolic=70,
                   triceps=29.2, insulin=160, bmi=32.4,
                   diabetes=0.6, age=34)
```

Working on the response scale, we use `predict` to get a prediction and standard error:

```
pred <- predict(Model3, newdata = newx, type = "response", se.fit = TRUE)
pred
## $fit
##         1
## 0.1901944
##
## $se.fit
##          1
## 0.02977891
##
## $residual.scale
## [1] 1
```

The 90% confidence interval based on approximate normality is

```
c(pred$fit + pred$se.fit * qnorm(0.05),
  pred$fit + pred$se.fit * qnorm(0.95))
##         1         1
## 0.1412124 0.2391763
```

Working on the link scale, we do

```
logistic <- function(x) { exp(x) / (1 + exp(x)) }
pred <- predict(Model3, newdata = newx, type = "link", se.fit = TRUE)
logistic(pred$fit)
##         1
## 0.1901944
logistic(c(pred$fit + pred$se.fit * qnorm(0.05),
           pred$fit + pred$se.fit * qnorm(0.95)))
##         1         1
## 0.1459445 0.2440272
```

Notice that the point prediction is the same either way, but the confidence intervals differ. The reason that the link scale is better is that the linear predictor itself is asymptotically normal, whereas the prediction on the response scale is only approximately asymptotically normal, since it's a smooth (logistic) transformation of an asymptotically normal quantity. The difference is, however, quite small in this case.

(h) Consider a Pima woman whose predictors are all the same as those of the woman in part (g) except that her `diabetes` pedigree function is 0.25. How much different are the log-odds of $Y = 1$ for these two women? Give a point estimate as well as a 90% confidence interval for the difference. Base all answers on the fitted Model 3.

**Solution:** Since we are comparing results for two potential subjects whose predictor vectors differ in one coordinate only, the difference between their log-odds of success ($\log(p/(1-p))$) will just equal the fitted coefficient for the one coordinate that changes times the difference between the two coordinate values. That is, let

$$x_1 = (x_{1,1}, \ldots, x_{1,p}), \text{ and}$$
$$x_2 = (x_{2,1}, \ldots, x_{2,p}),$$

be the predictor vectors for the two potential subjects. We have $x_{1,s} = x_{2,s}$ for all $s$ except for one value, say $s = j$. Then the difference between the subjects' log-odds will be $\beta_j[x_{2,j} - x_{1,j}]$. In our case $j$ corresponds to `diabetes`, while

$$x_{2,j} - x_{1,j} = 0.25 - 0.6 = -0.35,$$

and $\hat{\beta}_j = 1.151$. The standard error of $\hat{\beta}_j[x_{2,j} - x_{1,j}]$ is just the standard error of $\hat{\beta}_j$ times $|x_{2,j} - x_{1,j}|$. The standard error can then be combined with the 0.05 and 0.95 quantiles from the normal distribution to produce the asymptotic confidence interval. Here are the calculations:

```
est.diff <- coef(Model3)["diabetes"] * -0.35
est.diff
##   diabetes
## -0.4028195
## Standard error
std.err <- summary(Model3)$coefficients["diabetes", 2] * 0.35
std.err
## [1] 0.1484848
## 90 confidence interval
c(est.diff + std.err * qnorm(0.05),
  est.diff + std.err * qnorm(0.95))
##   diabetes   diabetes
## -0.6470553 -0.1585837
```

(i) The tests in parts (c) and (e) give us some idea about how much better one model fits than another between the three models involved. But does any of them fit well or are they all garbage? One way to get at this question is by grouping observations by their estimates of $\mathbb{P}(Y = 1 \mid X)$ and seeing whether the fractions of $Y = 1$ cases are close to the estimated probabilities.

For example, take all the cases where the logistic regression's estimated $\mathbb{P}(Y = 1 \mid X)$ is between 0.9 and 1; if the actual percentage of these with $Y = 1$ is substantially lower than 90%, that indicates a problem.

A model is called *well-calibrated* if the estimated probabilities $\mathbb{P}(Y = 1 \mid X)$ are close to the true fractions of $Y = 1$ cases for different values of $X$. Model 2 is trivially well-calibrated, since it has only one estimated probability for all observations, and that estimated probability equals the fraction of $Y = 1$ cases. Verify that claim based on your earlier analysis.

Models 1 and 3 have 392 different estimated probabilities spread from 0 to 1. We can get these probabilities using the `fitted` function in R. Checking how well-calibrated these models are can be thought of as a smoothing problem. For example, consider using a kernel smother to predict `test` using the estimated probabilities. For a specific predicted probability, say 0.8, the kernel regression fit will be a weighted average of the `test` values of points nearby—and the average of

0/1 observations gives the fraction that are 1. Hence the smoother will give (approximately) the correspondence between the predicted probabilities and the true rate of $Y = 1$.

Use a kernel regression to smooth `test` with `fitted(Model3)` as the predictor. (Let `npreg` use bandwidth 0.075.)

Plot the kernel regression's fitted values against `fitted(Model3)` to see how well-calibrated Model 3 is. Ideally the plot should look like the line $y = x$. Of course it does not look like the line $y = x$. Comment on how well-calibrated Model 3 seems to be.

Do the same thing with Model 1. Does either of these models appear to be noticeably better calibrated than the other?

**Solution:** We verified that Model 2 is trivially well-calibrated in part (c). For the other models, we need to do the requested smoothing. First, Model 1:

```
suppressPackageStartupMessages(library(np))

kernel1 <- npreg(Pima$test ~ fitted(Model1), bws=0.075)
```

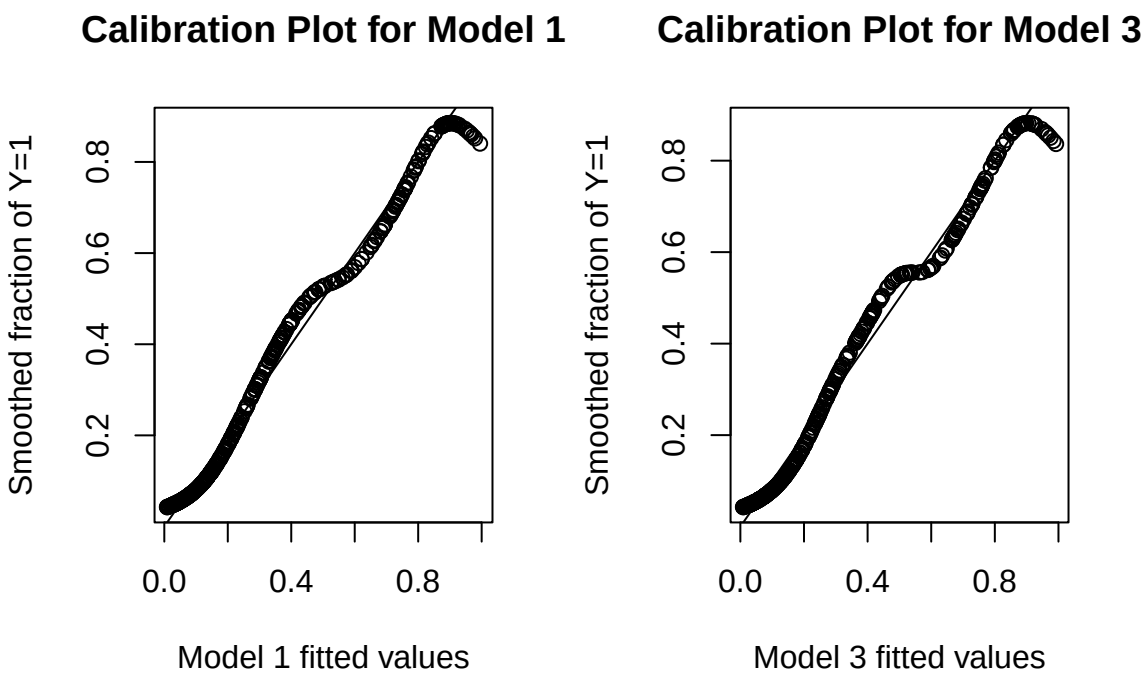
Next, we fit the same for Model 3:

```
kernel3 <- npreg(Pima$test ~ fitted(Model3), bws=0.075)
```

Let's plot both and compare their calibration.

```
par(mfrow=c(1,2))
plot(fitted(Model1), fitted(kernel1),
     xlab="Model 1 fitted values", main="Calibration Plot for Model 1",
     ylab="Smoothed fraction of Y=1")
abline(0, 1)

plot(fitted(Model3), fitted(kernel3),
     xlab="Model 3 fitted values", main="Calibration Plot for Model 3",
     ylab="Smoothed fraction of Y=1")
abline(0, 1)
```




The two plots are similar. They both have a region near fitted values of 0.5 where the fractions of $Y = 1$ seem too high for smaller fitted values and too low for higher fitted values. The Model1 plot is a little better in this middle region.

We can do something a little more quantitative by simply comparing the average distances between the smooth curves and the $y = x$ lines in the two plots. To avoid the end effects, we will compare only from the 35th smallest fitted value to the 35th largest.

For Model 1:

```
themiddle1 <- (fitted(Model1) >= sort(fitted(Model1))[35]) &
    (fitted(Model1) <= sort(fitted(Model1))[n - 34])

mean(abs(fitted(Model1)[themiddle1] - fitted(kernel1)[themiddle1]))
## [1] 0.02209323
```

For Model 3:

```
themiddle3 <- (fitted(Model3) >= sort(fitted(Model3))[35]) &
    (fitted(Model3) <= sort(fitted(Model3))[n - 34])

mean(abs(fitted(Model3)[themiddle3] - fitted(kernel3)[themiddle3]))
## [1] 0.02394233
```

The Model 1 smooth is closer, on average, to the corresponding fitted value, which is what is located on the $y = x$ line. Hence, Model 1 seems to be a little better calibrated than Model 3.

2. **Data Exam Two preparation.** You don't need to turn in anything for this problem. But you should do the following:

   - If there were any technology problems that slowed you down during the first data exam—problems getting R to work, issues knitting to PDF, anything like that—spend an hour to try to solve them

now.

- Review the rubric and sample exams on Canvas.
- If you anticipate any conflicts that will prevent you from completing the Data Exam during the week from April 29 to May 6, email professors Reinhart and Lee as soon as you know. (Please send the email to both of us, so we can coordinate with each other.)

You will receive your graded Data Exam One back before the second exam starts, and as soon as you do, review it alongside the rubric. Ask us if you have questions about your grade or how you can improve your report.

3. **Celebrate.** In celebration of the last 402 homework assignment this semester, we encourage you to submit a statistics meme, pet photo, funny quote, or other [class-appropriate] item.