432 Class 07

https://thomaselove.github.io/432-2024/

2024-02-06

## Today’s Agenda

* A First Example: Space Shuttle O-Rings
* Predicting a Binary outcome using a single predictor
  + using a linear probability model
  + using logistic regression and glm
  + using logistic regression and lrm

See Chapters 19-21 in our Course Notes for more on these models.

## Today’s R Setup

knitr::opts\_chunk$set(comment = NA)  
  
library(faraway) # data source  
library(broom)  
library(gt)  
library(patchwork)  
library(rms)  
library(tidyverse)  
  
theme\_set(theme\_bw())

## Challenger Space Shuttle Data

The US space shuttle Challenger exploded on 1986-01-28. An investigation ensued into the reliability of the shuttle’s propulsion system. The explosion was eventually traced to the failure of one of the three field joints on one of the two solid booster rockets. Each of these six field joints includes two O-rings which can fail.

* The discussion among engineers and managers raised concern that the probability of failure of the O-rings depended on the temperature at launch, which was forecast to be 31 degrees F.
* There are strong engineering reasons based on the composition of O-rings to support the judgment that failure probability may rise monotonically as temperature drops.

We have data on 23 space shuttle flights that preceded *Challenger* on primary O-ring erosion and/or blowby and on the temperature in degrees Fahrenheit. No previous liftoff temperature was under 53 degrees F.

## The “O-rings” data

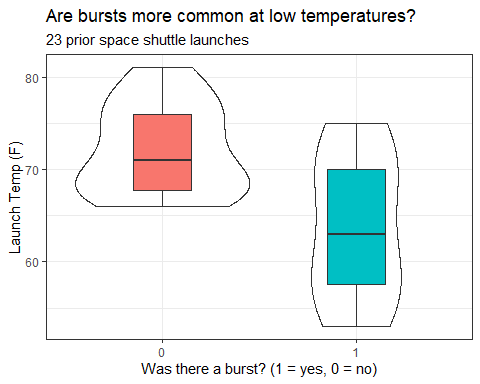
* damage = number of damage incidents out of 6 possible
* we set burst = 1 if damage > 0

orings1 <- faraway::orings |> tibble() |>  
 mutate(burst = case\_when( damage > 0 ~ 1, TRUE ~ 0))  
  
orings1 |> summary()

temp damage burst   
 Min. :53.00 Min. :0.0000 Min. :0.0000   
 1st Qu.:67.00 1st Qu.:0.0000 1st Qu.:0.0000   
 Median :70.00 Median :0.0000 Median :0.0000   
 Mean :69.57 Mean :0.4783 Mean :0.3043   
 3rd Qu.:75.00 3rd Qu.:1.0000 3rd Qu.:1.0000   
 Max. :81.00 Max. :5.0000 Max. :1.0000

## Association of burst and temp

ggplot(orings1, aes(x = factor(burst), y = temp)) +  
 geom\_violin() +   
 geom\_boxplot(aes(fill = factor(burst)), width = 0.3) +  
 guides(fill = "none") +   
 labs(title = "Are bursts more common at low temperatures?",  
 subtitle = "23 prior space shuttle launches",  
 x = "Was there a burst? (1 = yes, 0 = no)",   
 y = "Launch Temp (F)")

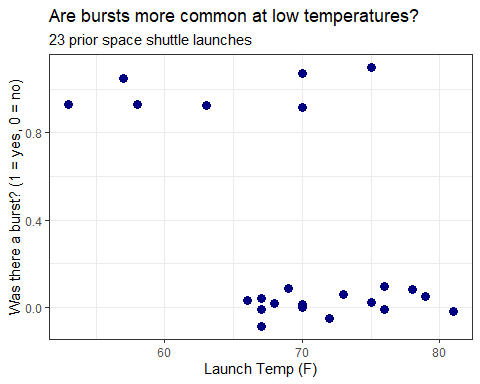


## Predict Prob(burst) using temperature?

We want to treat the binary variable burst as the outcome, and temp as the predictor.

* We’ll jitter the points vertically so that they don’t overlap completely if we have two launches with the same temperature.

ggplot(orings1, aes(x = temp, y = burst)) +  
 geom\_jitter(col = "navy", size = 3, width = 0, height = 0.1) +  
 labs(title = "Are bursts more common at low temperatures?",  
 subtitle = "23 prior space shuttle launches",  
 y = "Was there a burst? (1 = yes, 0 = no)",   
 x = "Launch Temp (F)")



# A Linear Probability Model, fit with lm()

## Linear model to predict Prob(burst)?

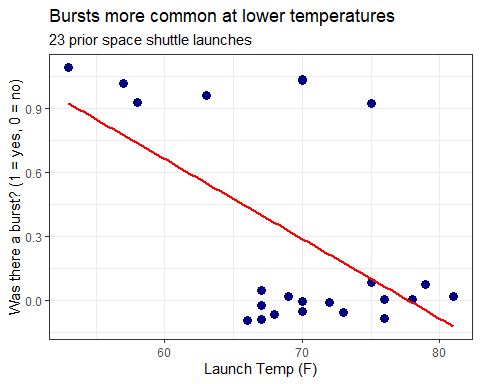
mod1 <- lm(burst ~ temp, data = orings1)  
  
tidy(mod1, conf.int = T) |> gt() |>  
 fmt\_number(decimals = 3) |> tab\_options(table.font.size = 20)

| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
| --- | --- | --- | --- | --- | --- | --- |
| (Intercept) | 2.905 | 0.842 | 3.450 | 0.002 | 1.154 | 4.656 |
| temp | -0.037 | 0.012 | -3.103 | 0.005 | -0.062 | -0.012 |

* This is a **linear probability model**.

## Plot linear probability model?

ggplot(orings1, aes(x = temp, y = burst)) +  
 geom\_jitter(col = "navy", size = 3, width = 0, height = 0.1) +  
 geom\_smooth(method = "lm", se = F, col = "red",  
 formula = y ~ x) +  
 labs(title = "Bursts more common at lower temperatures",  
 subtitle = "23 prior space shuttle launches",  
 y = "Was there a burst? (1 = yes, 0 = no)",   
 x = "Launch Temp (F)")



* It would help if we could see the individual launches…

## Making Predictions with mod1

mod1$coefficients

(Intercept) temp   
 2.90476190 -0.03738095

* What does mod1 predict for the probability of a burst if the temperature at launch is 70 degrees F?

predict(mod1, newdata = tibble(temp = 70))

1   
0.2880952

* What if the temperature was actually 60 degrees F?

## Making Several Predictions with mod1

Let’s use our linear probability model mod1 to predict the probability of a burst at some other temperatures…

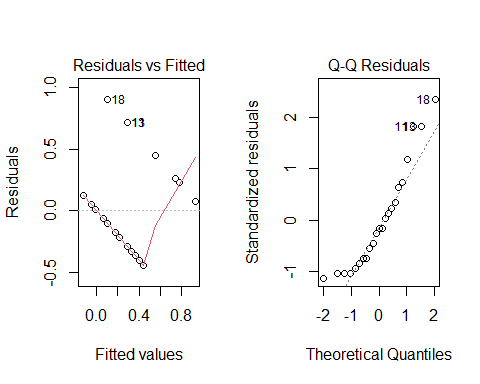
newtemps <- tibble(temp = c(80, 70, 60, 50, 31))  
  
augment(mod1, newdata = newtemps)

# A tibble: 5 × 2  
 temp .fitted  
 <dbl> <dbl>  
1 80 -0.0857  
2 70 0.288   
3 60 0.662   
4 50 1.04   
5 31 1.75

* Uh, oh.

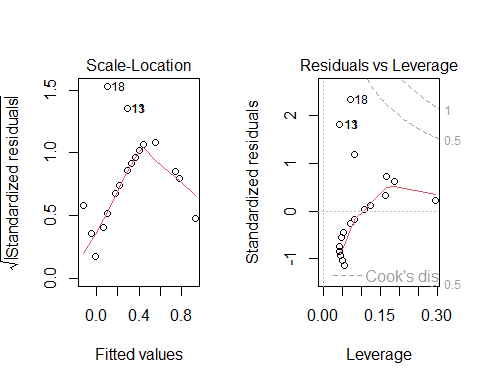
## Residual Plots for mod1? (1/2)

par(mfrow = c(1,2)); plot(mod1, which = c(1:2))



## Residual Plots for mod1? (2/2)

par(mfrow = c(1,2)); plot(mod1, which = c(3,5))



## Models to predict a Binary Outcome

Our outcome takes on two values (zero or one) and we then model the probability of a “one” response given a linear function of predictors.

Idea 1: Use a *linear probability model*

* Main problem: predicted probabilities that are less than 0 and/or greater than 1
* Also, how can we assume Normally distributed residuals when outcomes are 1 or 0?

## Models to predict a Binary Outcome

Idea 2: Build a *non-linear* regression approach

* Most common approach: logistic regression, part of the class of *generalized* linear models

# A Logistic Regression Model, fit with glm()

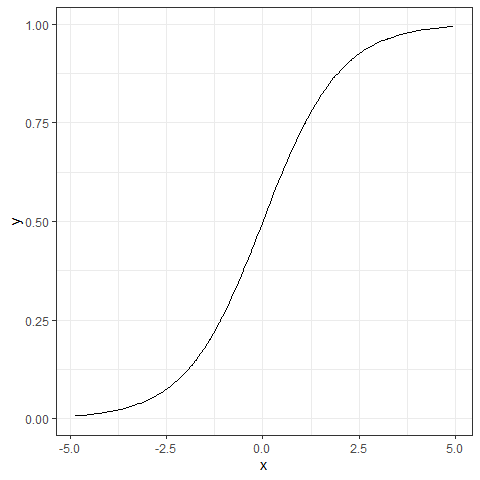
## The Logit Link and Logistic Function

The function we use in logistic regression is called the **logit link**.

The inverse of the logit function is called the **logistic function**. If logit() = , then .

* The logistic function takes any value in the real numbers and returns a value between 0 and 1.

## The Logistic Function



## The logit or log odds

We usually focus on the **logit** in statistical work, which is the inverse of the logistic function.

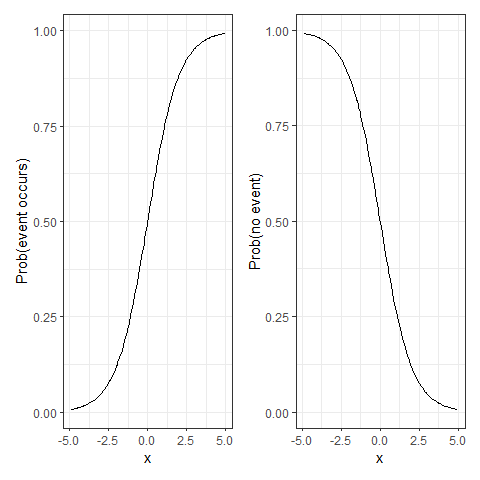
* If we have a probability , then .
* If our probability , then .
* Finally, if , then .

### Why is this helpful?

* log(odds(Y = 1)) or logit(Y = 1) covers all real numbers.
* Prob(Y = 1) is restricted to [0, 1].

## Predicting Pr(event) or Pr(no event)

* Can we flip the story?



## Back to predicting Prob(burst)

We’ll use the glm function in R, specifying a logistic regression model.

* Instead of predicting , we’re predicting or .

## mod2 for Prob(burst)

mod2 <- glm(burst ~ temp, data = orings1,  
 family = binomial(link = "logit"))  
  
tidy(mod2, conf.int = TRUE) |> gt() |>  
 fmt\_number(decimals = 3) |> tab\_options(table.font.size = 20)

| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
| --- | --- | --- | --- | --- | --- | --- |
| (Intercept) | 15.043 | 7.379 | 2.039 | 0.041 | 3.331 | 34.342 |
| temp | -0.232 | 0.108 | -2.145 | 0.032 | -0.515 | -0.061 |

## Predicting with mod2

* For a temperature of 70 F at launch, what is our prediction?

log(odds(burst)) = 15.043 - 0.232 (70) = -1.197

* Exponentiate to get the odds…

odds(burst) = exp(-1.197) = 0.302

* so, we can estimate the probability by

## Prediction from mod2 for temp = 60

What is the predicted probability of a burst if the temperature is 60 degrees?

* log(odds(burst)) = 15.043 - 0.232 (60) = 1.123
* odds(burst) = exp(1.123) = 3.074
* Pr(burst) = 3.074 / (3.074 + 1) = 0.755

## Will augment do this, as well?

Yes, and it will retain many more decimal places in intermediate calculations…

temps <- tibble(temp = c(60,70))  
  
augment(mod2, newdata = temps, type.predict = "link")

# A tibble: 2 × 2  
 temp .fitted  
 <dbl> <dbl>  
1 60 1.11  
2 70 -1.21

augment(mod2, newdata = temps, type.predict = "response")

# A tibble: 2 × 2  
 temp .fitted  
 <dbl> <dbl>  
1 60 0.753  
2 70 0.230

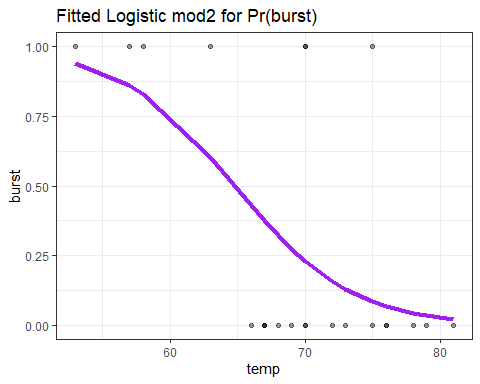
## Plotting the Logistic Regression Model

Use the augment function to get the fitted probabilities into the original data, then plot.

* Note that we’re just connecting the predictions made for observed temp values with geom\_line, so the appearance of the function isn’t as smooth as the actual logistic regression model.

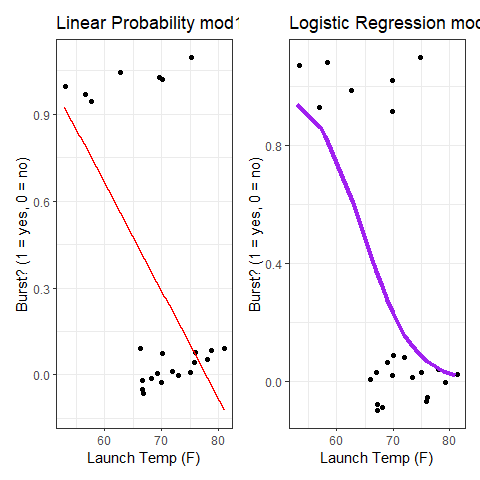
mod2\_aug <- augment(mod2, type.predict = "response")  
  
ggplot(mod2\_aug, aes(x = temp, y = burst)) +  
 geom\_point(alpha = 0.4) +  
 geom\_line(aes(x = temp, y = .fitted),   
 col = "purple", size = 1.5) +  
 labs(title = "Fitted Logistic mod2 for Pr(burst)")

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
ℹ Please use `linewidth` instead.



## Comparing fits of mod1 and mod2

p1 <- ggplot(orings1, aes(x = temp, y = burst)) +  
 geom\_jitter(height = 0.1) +  
 geom\_smooth(method = "lm", se = F, col = "red",  
 formula = y ~ x) +  
 labs(title = "Linear Probability mod1",  
 y = "Burst? (1 = yes, 0 = no)",   
 x = "Launch Temp (F)")  
  
  
p2 <- ggplot(mod2\_aug, aes(x = temp, y = burst)) +  
 geom\_jitter(height = 0.1) +  
 geom\_line(aes(x = temp, y = .fitted),   
 col = "purple", size = 1.5) +  
 labs(title = "Logistic Regression mod2",  
 y = "Burst? (1 = yes, 0 = no)",   
 x = "Launch Temp (F)")  
  
p1 + p2



## Try exponentiating mod2 coefficients?

How can we interpret the coefficients of the model?

### Exponentiating the slope is helpful

exp(-0.232)

[1] 0.7929461

## Exponentiating the slope helps

exp(-0.232)

[1] 0.7929461

Suppose Launch A’s temperature was one degree higher than Launch B’s.

* The **odds** of Launch A having a burst are 0.793 times as large as they are for Launch B.
* Odds Ratio estimate comparing two launches whose temp differs by 1 degree is 0.793

## Exponentiated and tidied slope of temp (mod2)

tidy(mod2, exponentiate = TRUE, conf.int = TRUE) |>  
 filter(term == "temp") |>  
 gt() |> fmt\_number(decimals = 3) |>   
 tab\_options(table.font.size = 20)

| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
| --- | --- | --- | --- | --- | --- | --- |
| temp | 0.793 | 0.108 | -2.145 | 0.032 | 0.597 | 0.941 |

* What would it mean if the Odds Ratio for temp was 1?
* How about an odds ratio that was greater than 1?

# A logistic regression model, fit with lrm() from **rms**

## Fitting the model again

d <- datadist(orings1)  
options(datadist = "d")  
  
mod3 <- lrm(burst ~ temp, data = orings1, x = TRUE, y = TRUE)

as compared to

mod2 <- glm(burst ~ temp, data = orings1,   
 family =binomial(link ="logit"))

These will fit the same model.

## mod3 Results

mod3

Logistic Regression Model  
  
lrm(formula = burst ~ temp, data = orings1, x = TRUE, y = TRUE)  
  
 Model Likelihood Discrimination Rank Discrim.   
 Ratio Test Indexes Indexes   
Obs 23 LR chi2 7.95 R2 0.413 C 0.781   
 0 16 d.f. 1 R2(1,23) 0.261 Dxy 0.562   
 1 7 Pr(> chi2) 0.0048 R2(1,14.6)0.379 gamma 0.589   
max |deriv| 0.0002 Brier 0.139 tau-a 0.249   
  
 Coef S.E. Wald Z Pr(>|Z|)  
Intercept 15.0429 7.3786 2.04 0.0415   
temp -0.2322 0.1082 -2.14 0.0320

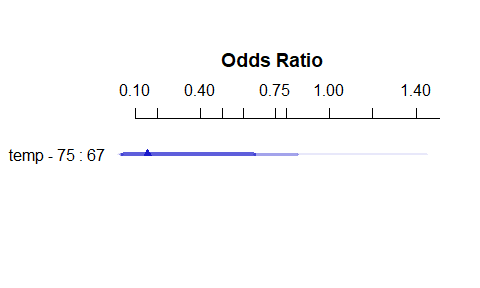
## summary(mod3) Results

summary(mod3)

Effects Response : burst   
  
 Factor Low High Diff. Effect S.E. Lower 0.95 Upper 0.95  
 temp 67 75 8 -1.85730 0.86589 -3.554400 -0.16018   
 Odds Ratio 67 75 8 0.15609 NA 0.028598 0.85199

### Effects Plot

plot(summary(mod3))



## Predictions from mod3

newdat <- tibble(temp = c(50, 60, 70, 80))  
  
## predictions on the log odds scale  
predict(mod3, newdata = newdat)

1 2 3 4   
 3.434751 1.113132 -1.208488 -3.530108

## predictions on the probability scale  
predict(mod3, newdata = newdat, type = c("fitted"))

1 2 3 4   
0.9687731 0.7527125 0.2299686 0.0284676

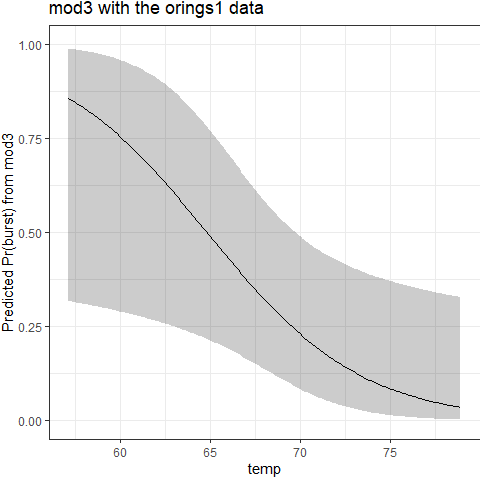
## Plot in-sample predictions on log-odds scale

ggplot(Predict(mod3))



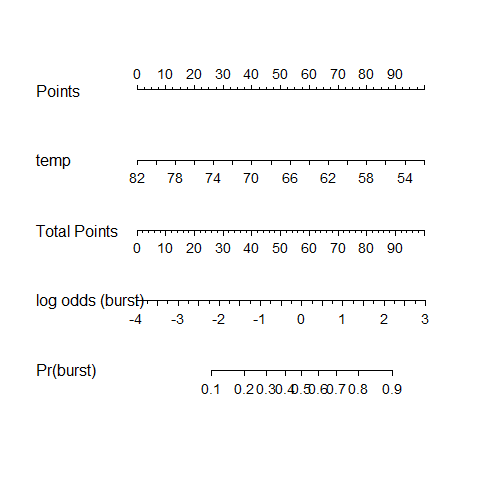
## Plot in-sample predictions on probability scale

ggplot(Predict(mod3, fun = plogis)) +  
 labs(y = "Predicted Pr(burst) from mod3",  
 title = "mod3 with the orings1 data")



## Nomogram for mod3

plot(nomogram(mod3, fun = plogis, funlabel = "Pr(burst)"),   
 lplabel="log odds (burst)")



## Regression on a Binary Outcome

**Linear Probability Model** (a linear model)

lm(event ~ predictor1 + predictor2 + ..., data = tibblename)

* Pr(event) is linear in the predictors

**Logistic Regression Model** (generalized linear model)

glm(event ~ pred1 + pred2 + ..., data = tibblename,  
 family = binomial(link = "logit"))  
or   
  
dd <- datadist(tibblename); options(datadist = "dd")  
lrm(event ~ pred1 + pred2 + ..., data = tibblename,   
 x = TRUE, y = TRUE)

* Logistic Regression forces a prediction in (0, 1)
* log(odds(event)) is linear in the predictors

## The logistic regression model

## Next Time

* Binary regression models with multiple predictors
* Assessing the quality of fit for a logistic model