Stat 340: Intro. to Classification and Logistic Regression

Example: Challenger Space Shuttle O-Rings

On January 28, 1986, the American space shuttle Challenger exploded 73 seconds into flight; all seven crew members on board died. It was later determined that the cause of the explosion was a failure in a joint in one of the booster rockets that launched the shuttle. The failure was due to damage to an O-ring that was used to seal the joint.

Can we predict probability of damage to an O-ring given the temperature on the morning of the launch?

$$
\begin{aligned}
Y_i &= \begin{cases} 1 & \text{if there was evidence of damage to an O-ring on launch number } i \\ 0 & \text{otherwise} \end{cases} \\
X_i &= \text{ temperature at launch for launch number } i
\end{aligned}
$$

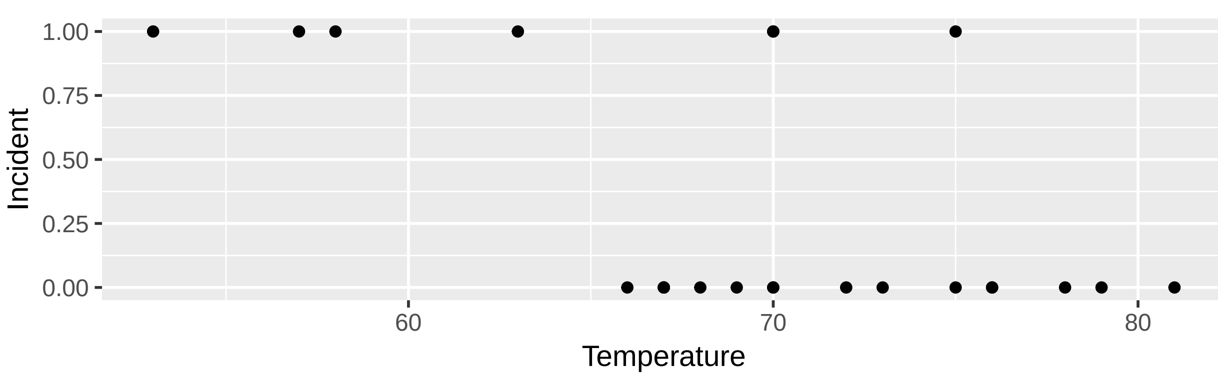
```
library(tidyverse)
challenger <- read_csv("http://www.evanlray.com/data/chihara_hesterberg/Challenger.csv")
head(challenger)
```

```
## # A tibble: 6 x 3
##   Date     Temperature Incident
##   <chr>          <int>    <int>
## 1 Apr12.81          66        0
## 2 Nov12.81          70        1
## 3 Mar22.82          69        0
## 4 Nov11.82          68        0
## 5 Apr04.83          67        0
## 6 Jun18.83          72        0
```

```
nrow(challenger)
```

```
## [1] 23
```

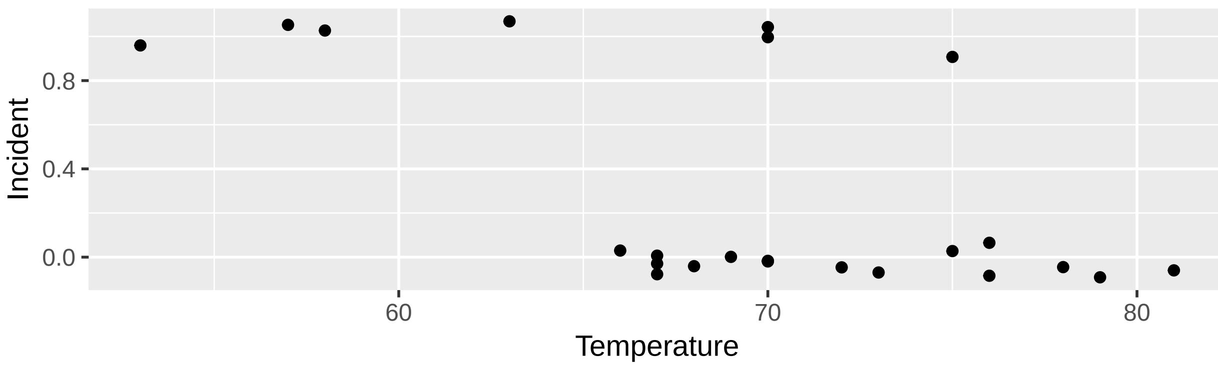
```
ggplot(data = challenger, mapping = aes(x = Temperature, y = Incident)) +
  geom_point()
```



Dealing with Overplotting

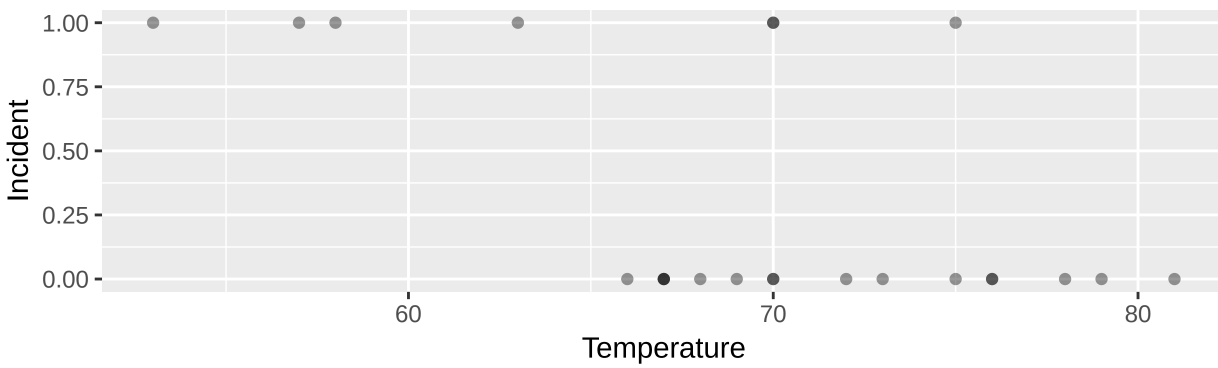
Setting jitter…

```
ggplot(data = challenger, mapping = aes(x = Temperature, y = Incident)) +
  geom_point(position = position_jitter(width = 0, height = 0.1))
```



Setting alpha…

```
ggplot(data = challenger, mapping = aes(x = Temperature, y = Incident)) +
  geom_point(alpha = 0.4)
```

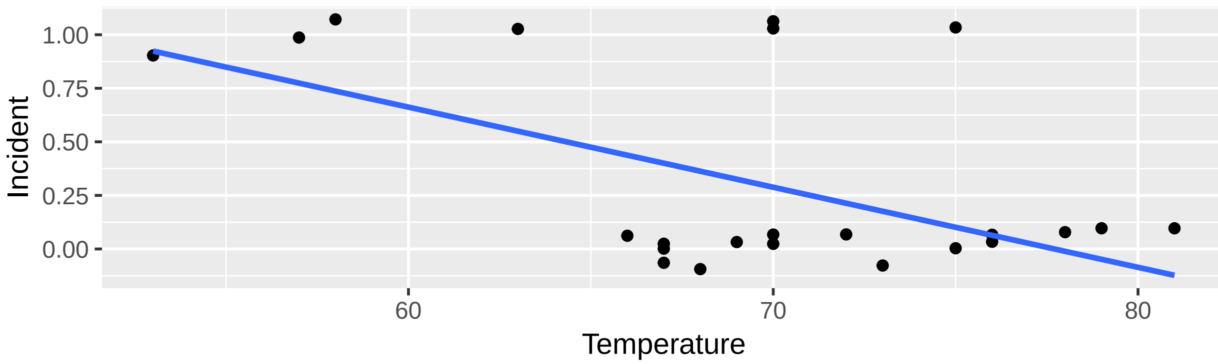


Analysis via Simple Linear Regression

Suppose we use the model

$P(Y_i = 1|X_i) = p(X_i) = \beta_0 + \beta_1 X_i$

```r
ggplot(data = challenger, mapping = aes(x = Temperature, y = Incident)) +
  geom_point(position = position_jitter(width = 0, height = 0.1)) +
  geom_smooth(method = "lm", se = FALSE)
```



Is this ok?

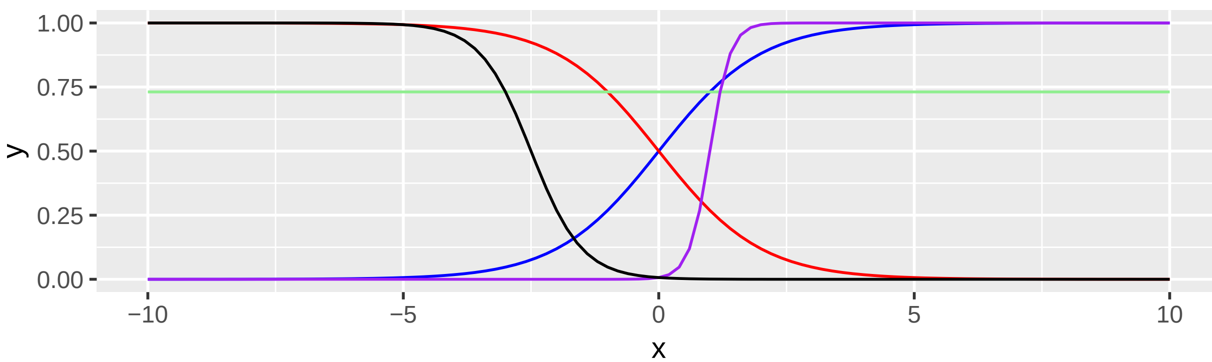
Alternative: Logistic Regression

$$P(Y_i = 1|X_i) = p(X_i) = \frac{e^{\beta_0 + \beta_1 X_i}}{1 + e^{\beta_0 + \beta_1 X_i}}$$

This function is called the **logistic function**.

```
#' Evaluate the logistic regression function
#'
#' @param x a vector of values at which to evaluate the logistic regression function
#' @param beta_0 value for the intercept parameter of a logistic regression model
#' @param beta_1 value for the slope parameter of a logistic regression model
#'
#' @return vector of predicted probabilities that Y = 1 given x
logistic <- function(x, beta_0, beta_1) {
  return(exp(beta_0 + beta_1 * x) / (1 + exp(beta_0 + beta_1 * x)))
}

ggplot(mapping = aes(x = c(-10, 10))) +
  stat_function(fun = logistic, args = list(beta_0 = 0, beta_1 = 1), color = "blue") +
  stat_function(fun = logistic, args = list(beta_0 = 0, beta_1 = -1), color = "red") +
  stat_function(fun = logistic, args = list(beta_0 = 1, beta_1 = 0), color = "lightgreen") +
  stat_function(fun = logistic, args = list(beta_0 = -5, beta_1 = 5), color = "purple") +
  stat_function(fun = logistic, args = list(beta_0 = -5, beta_1 = -2), color = "black") +
  xlab("x")
```



Observations:

- For all possible values of X_i, $P(Y_i = 1|X_i) \in (0, 1)$
- β_1 controls direction of curve:
  - if $\beta_1 > 0$, then $p(x)$ is increasing in x
  - if $\beta_1 < 0$, then $p(x)$ is decreasing in x
  - if $\beta_1 = 0$, then $p(x)$ does not depend on the value of x.
- β_1 also controls "slope" of curve:
  - if $|\beta_1|$ is large, then $p(x)$ changes between 0 and 1 quickly
  - if $|\beta_1|$ is small, then $p(x)$ changes between 0 and 1 slowly
  - The maximum slope is $\beta_1/4$, and occurs at the value of x where $p(x) = 0.5$
- β_0 shifts the curve left and right, but is otherwise not interpretable.

Applied to O-Rings Data:

- For estimation, similar to `lm` for fitting linear models, but…
  - Instead of `lm`, use `glm` (stands for generalized linear model)
  - Specify `family = binomial`

```
fit <- glm(Incident ~ Temperature, data = challenger, family = binomial)
summary(fit)
```

```
##
## Call:
## glm(formula = Incident ~ Temperature, family = binomial, data = challenger)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0611  -0.7613  -0.3783   0.4524   2.2175
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  15.0429     7.3786   2.039   0.0415 *
## Temperature  -0.2322     0.1082  -2.145   0.0320 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28.267  on 22  degrees of freedom
## Residual deviance: 20.315  on 21  degrees of freedom
## AIC: 24.315
##
## Number of Fisher Scoring iterations: 5
```

- For prediction based on fitted model, similar to use of `predict` with linear model fits, but…
  - Need to supply `type = "response"` to say that we want a prediction on the scale of the response variable

On the day of the Challenger explosion, the temperature was 33 degrees F. What does this model fit predict for probability of O-ring failure?

```
predict(fit, newdata = data.frame(Temperature = 33), type = "response")
```

```
##         1
## 0.9993777
```

Compare to:

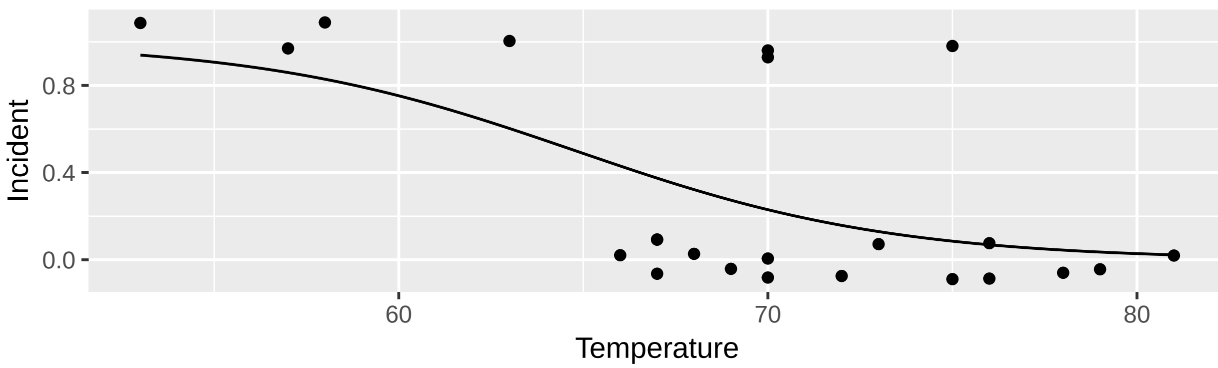
```
exp(15.0429 - 0.2322 * 33) / (1 + exp(15.0429 - 0.2322 * 33))
```

```
## [1] 0.999377
```

Here's a plot using `stat_function` to plot predicted values (we have to define the function that generates predictions).

```
#' Calculate predictions based on the logistic regression fit called "fit"
#' (obtained on page 5)
#'
#' @param x a vector of values for Temperature at which we want to make predictions
#'
#' @return a vector of estimated probabilitities that there will be O-ring damage
pred_logistic <- function(x) {
  predict(fit, newdata = data.frame(Temperature = x), type = "response")
}

ggplot(data = challenger, mapping = aes(x = Temperature, y = Incident)) +
  geom_point(position = position_jitter(width = 0, height = 0.1)) +
  stat_function(fun = pred_logistic)
```



What is our classification boundary, when we think O-ring damage is more likely than not?

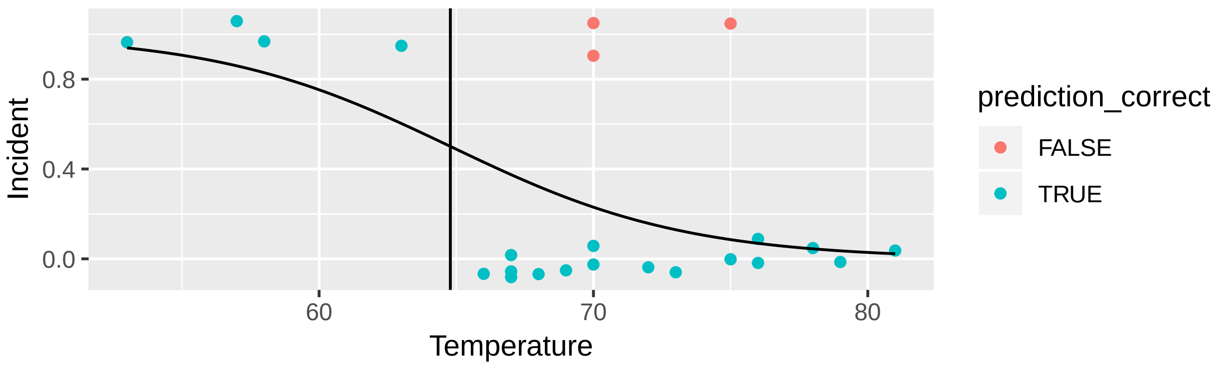
Set

$$P(Y_i = 1|X_i = x_i) = p(x_i) = \frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}} = 0.5$$

```
-15.0429 / (-0.2322)
```

```
## [1] 64.78424
```

```r
challenger <- challenger %>%
  mutate(
    predicted_prob_incident = predict(fit, type = "response"),
    predicted_incident = as.numeric(predicted_prob_incident > 0.5),
    prediction_correct = (Incident == predicted_incident)
  )
ggplot(data = challenger, mapping = aes(x = Temperature, y = Incident)) +
  geom_point(mapping = aes(color = prediction_correct),
    position = position_jitter(width = 0, height = 0.1)) +
  stat_function(fun = pred_logistic) +
  geom_vline(xintercept = -15.0429 / -0.2322)
```



What is our training error rate?

```r
challenger %>%
  summarize(
    prediction_error_rate = mean(Incident != predicted_incident)
  )
```

```
## # A tibble: 1 x 1
##   prediction_error_rate
##                   <dbl>
## 1                 0.130
```

Can also be calculated as. . .

```r
3/23
```

```
## [1] 0.1304348
```