CH 13: Logistic Regression

Motivation

Let's assume that we have access to the underlying candy face off data.

Consider the following model:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

Q: What issues might we have with this model?

Q: What are some possible solutions?

Logistic regression is a special case of

Logistic Regression

The logistic function maps an input from the unit range (0,1) to the real line:

$$logit(x) = \log\left(\frac{x}{1-x}\right)$$

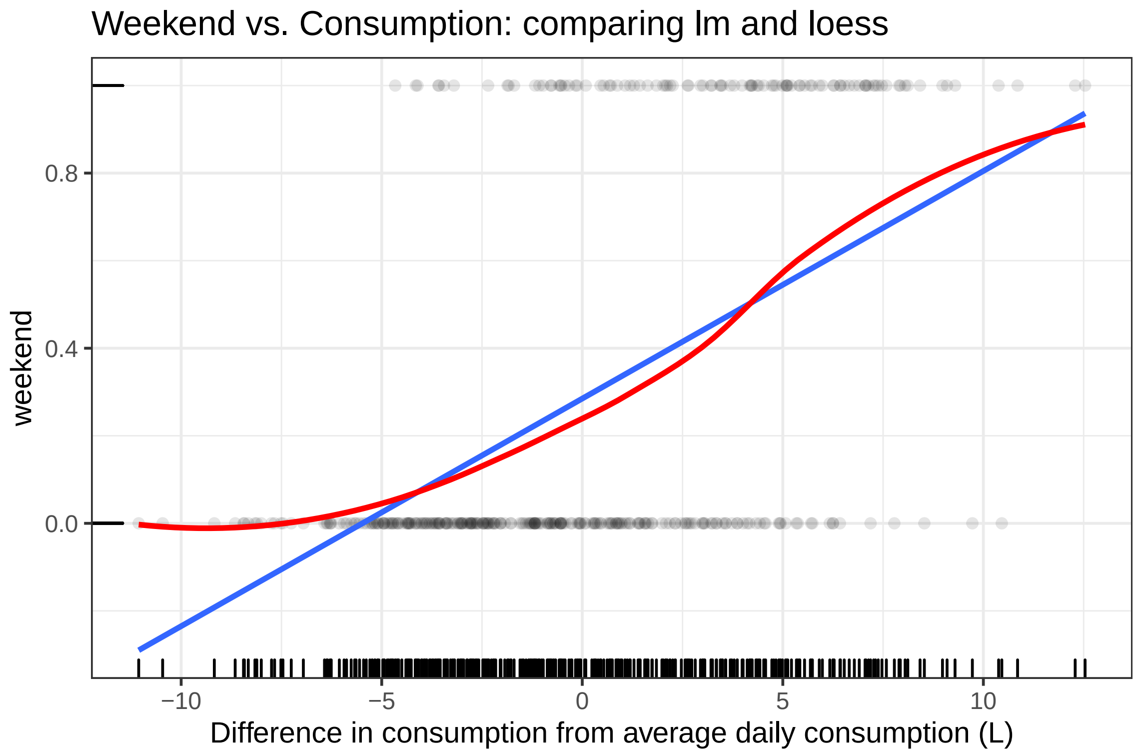
The `qlogis` (for logit) and `plogis` (inverse-logit) functions in R can be used for this calculation. For instance `plogis(1) =` 0.7310586.

Formally, the inverse-logistic function is used as part of the GLM:

Recall the `beer` dataset, but now instead of trying to model consumption, lets consider whether a day is a weekday or weekend.

```
beer <- read_csv('http://math.montana.edu/ahoegh/Data/Brazil_cerveja.csv') %>% mutate(consumed = consume
```

```
beer %>% ggplot(aes(y = weekend, x = consumed)) +
  geom_point(alpha = .1) +
  geom_smooth(formula = 'y~x', method = 'lm', se =F) +
  geom_smooth(formula = 'y~x', method = 'loess', color = 'red', se = F) +
  geom_rug() + ggtitle('Weekend vs. Consumption: comparing lm and loess') +
  theme_bw() + xlab('Difference in consumption from average daily consumption (L)')
```




```
bayes_logistic <- stan_glm(weekend ~ consumed, data = beer,
                           family = binomial(link = "logit"), refresh = 0)
```

```
freq_logistic <- glm(weekend ~ consumed, data = beer,
                     family = binomial(link = "logit"))
```

Now how to interpret the model coefficients?

```r
bayes_logistic
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      weekend ~ consumed
##  observations: 365
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -1.2    0.2
## consumed     0.3    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```r
summary(freq_logistic)
```

```
##
## Call:
## glm(formula = weekend ~ consumed, family = binomial(link = "logit"),
##     data = beer)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0968  -0.6859  -0.4178   0.7367   2.3624
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.24466    0.15059  -8.265   <2e-16 ***
## consumed     0.31791    0.03773   8.427   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 436.21  on 364  degrees of freedom
## Residual deviance: 333.74  on 363  degrees of freedom
## AIC: 337.74
##
## Number of Fisher Scoring iterations: 5
```

Interpreting the coefficients can be challenging due to the non-linear relationship between the outcome and the predictors.

Predictive interpretation

One way to interpret the coefficients is in a predictive standpoint. For instance, consider an day with average consumption, then the probability of a weekend would be `invlogit(-1.2 + 0.3 * 0)` = 0.23, where as the probability of a day with 10 more liters of consumption (relative to an average day) would have a weekend probability of `invlogit(-1.2 + 0.3 * 10)` = 0.86

Of course, we should always think about uncertainty, so we can extract simulations from the model.

`posterior_linpred` was useful with regression

```r
new_data <- data.frame(consumed = c(0,10))
posterior_sims <- posterior_linpred(bayes_logistic, newdata = new_data)
summary(posterior_sims)
```

```
##        1                  2
##  Min.   :-1.9977   Min.   :0.7572
##  1st Qu.:-1.3556   1st Qu.:1.7180
##  Median :-1.2483   Median :1.9415
##  Mean   :-1.2517   Mean   :1.9431
##  3rd Qu.:-1.1480   3rd Qu.:2.1707
##  Max.   :-0.7413   Max.   :3.1478
```

```r
posterior_sims <- posterior_epred(bayes_logistic, newdata = new_data)
summary(posterior_sims)
```

```
##        1                 2
##  Min.   :0.1194   Min.   :0.6808
##  1st Qu.:0.2050   1st Qu.:0.8479
##  Median :0.2230   Median :0.8745
##  Mean   :0.2235   Mean   :0.8700
##  3rd Qu.:0.2408   3rd Qu.:0.8976
##  Max.   :0.3227   Max.   :0.9588
```

It can also be useful to consider predictions of an individual data point.

```
new_obs <- posterior_predict(bayes_logistic, newdata = new_data)
head(new_obs)
```

```
##      1 2
## [1,] 0 1
## [2,] 0 1
## [3,] 0 1
## [4,] 0 1
## [5,] 0 0
## [6,] 0 1
```

```
colMeans(new_obs)
```

```
##     1     2
## 0.228 0.876
```

Model Comparison

We can use cross validation in the same manner a standard linear models.

```
loo(bayes_logistic)
```

```
##
## Computed from 4000 by 365 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo   -168.9 10.5
## p_loo         2.0  0.2
## looic       337.7 20.9
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
temp_model <- stan_glm(weekend~max_tmp, data = beer, refresh=0)
loo(temp_model)
```

```
##
## Computed from 4000 by 365 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo   -230.1  9.1
## p_loo         2.5  0.2
## looic       460.1 18.2
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```