Cheat Sheet: Logistic Regression

and Poisson and Ordered Logistic Regression

Measurement and Evaluation of HCC Systems

Scenario

Use regression if you want to test the simultaneous linear effect of several variables `varX1`, `varX2`, … on a binary (logistic), count (Poisson) or ordinal (ordered logistic) outcome variable `varY`. In this scenario, you are predicting `varY` with `varX1`, `varX2`, …. Note that for the Poisson distribution to hold, counts should be decreasing in frequency (many zeros, fewer ones, even fewer twos, etc.).

Power analysis for logistic regression

- Power analysis for logistic regression is beyond the scope of this course.

Plotting scatterplots with linear trend line

- See the linear regression cheat sheet for creating scatterplots, and/or the *t* test and ANOVA cheats for creating bar charts, line charts, and box plots.
- (logistic regression only) If your *Y* variable is has nominal values (e.g. "A" and "B" instead of 0 and 1), you can turn it into a numeric variable by using `as.numeric(varY == "B")`.
- For error bars, use `mean_cl_boot` instead of `mean_cl_normal`.

Pre-testing assumptions

- Make sure that your Y is independent. Normality is not required here.

(optional) Preparing dummy variables

- This is exactly the same as linear regression.

Running the test

- Run the regression model as follows (include additional *X*s if needed):
  - Logistic regression:
    ```
    model1 <- glm(varY ~ varX1 + varX2, data = data, family = binomial)
    ```
  - Poisson regression:
    ```
    model1 <- glm(varY ~ varX1 + varX2, data = data, family = poisson)
    ```

- Ordered Logistic regression:
  `model1 <- polr(factor(varY) ~ varX1 + varX2, data = data, Hess=T)`
  For the latter, also run a null model:
  `model1.null <- polr(factor(varY) ~ 1, data = data, Hess=T)`

- You can test the improvement of the model with a χ^2-statistic (similar to the *F*-statistic in linear regression).
  - First calculate the likelihood ratio (for ordered logistic regression, use `model1.null$deviance` instead of `model1$null.deviance`):
    `ratio <- model1$null.deviance - model1$deviance`
  - Then calculate the degrees of freedom (for ordered logistic regression, use `model1.null$df.residual` instead of `model1$df.null`):
    `df <- model1$df.null - model1$df.residual`
  - Finally, calculate the *p*-value:
    `1-pchisq(ratio, df)`
- You can also calculate the model R^2; there are 3 versions (for ordered logistic regression, use `model1.null$deviance` instead of `model1$null.deviance`):
  - Hosmer-Lemeshow:
    `ratio / model1$null.deviance`
  - Cox-Snell:
    `1 - exp(-ratio / dim(data)[1])`
  - Nagelkerke:
    `(1 - exp(-ratio / dim(data)[1])) / (1 - exp(-model1$null.deviance / dim(data)[1]))`
- Get the model summary:
  `summary(model1)`
- Interpret the coefficients:
  - For logistic regression, the probability of *Y* is given as $P(Y) = 1/(1+e^{-(a+b1*varX1+b2*varX2+...)})$.
    e^b is the odds ratio in *Y* of a 1-point increase in *X*, or, if *X* is a dummy variable, the odds ratio in *Y* between this category and the baseline category.
  - For Poisson regression, the rate of *Y* is given as $Y = e^{a+b1*varX1+b2*varX2+...}$.
    e^b is the rate ratio in *Y* of a 1-point increase in *X*, or, if *X* is a dummy variable, the rate ratio in *Y* between this category and the baseline category.
  - For ordered logistic regression, the probability of *Y* being a certain value or higher is given as $P(Y) = 1/(1+e^{-(a+b1*varX1+b2*varX2+...)})$, where a consists for *k*–1 thresholds.
    e^b is the odds ratio in *Y* being a certain value or higher of a 1-point increase in *X*, or, if *X* is a dummy variable, the odds ratio in *Y* being a certain value or higher between this category and the baseline category.

- You can easily get e^b and its confidence interval:
  ```
  exp(model1$coefficients)
  exp(confint(model1))
  ```
- You can interpret e^b = 1.xx as an xx% increase in *Y* for each increase in *X*. You can interpret e^b = 0.xx as a (100-xx)% decrease in *Y* for each increase in *X*. You can interpret e^b = x.xx as an x.xx-fold increase in *Y* for each increase in *X*. If you want to get the effect on *Y* for a *decrease* in *X*, you can calculate $1/e^b$.

- Each coefficient has a *t* test and a *p*-value to test if the effect is significant. Multiply the *p*-value by 2 if you were conducting a one-sided test (i.e. if you had a directional hypothesis).

(optional) Robust versions

- Bootstrapping works the same as in linear regression, so refer to that cheat sheet.
- You can also use a sandwich estimator of the standard error using the package `sandwich`. This also works for regular linear regression!
  ```
  cov.model1 <- vcovHC(model1, type="HC0")
  std.err <- sqrt(diag(cov.model1))
  pval <- 2 * pnorm(abs(coef(model1)/std.err), lower.tail=F)
  LL <- coef(model1) - 1.96 * std.err
  UL <- coef(model1) + 1.96 * std.err
  ```
  Here, `std.err` is the robust standard error, `pval` is the robust *p*-value, and `LL` and `UL` are the lower and upper limits of the robust confidence interval.

(optional) Testing additional variables

- This is the same as linear regression, so refer to that cheat sheet.
- The only difference is that the ANOVA test of the difference between two models does not produce a *p*-value. This *p*-value can be calculated as follows:
  ```
  1-pchisq(model1$deviance - model2$deviance, model1$df.residual -
  model2$df.residual)
  ```

Post-testing assumptions and inspecting outliers

- This is exactly the same as for linear regression, so refer to that cheat sheet.

Reporting

- This is also the same as for linear regression, but with three differences:
  - Report not just *b* and SE_b, but also the odds ratio (e^b), and maybe its confidence interval.
  - Make sure to report an R^2 (Nagelkerke is most common) and the model χ^2.
  - Also report ΔR^2 and the χ^2 difference test if you compare multiple models.
  - In the textual explanation, report on your coefficients as odds ratios (e^b).