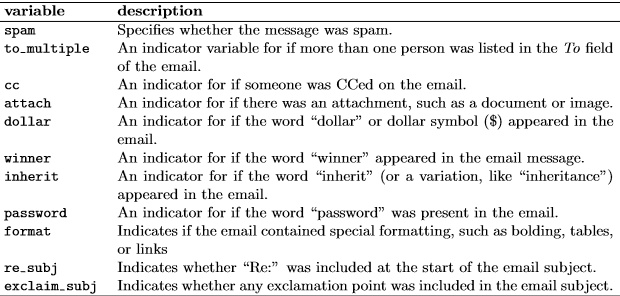
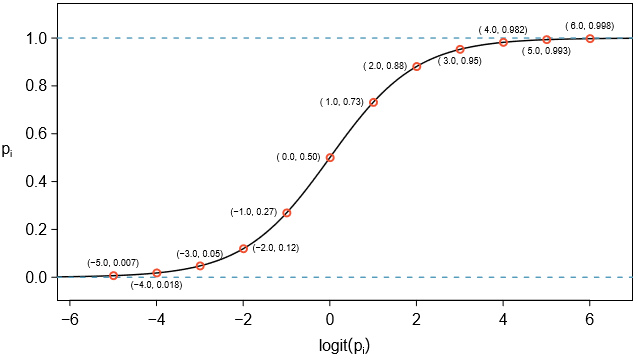
# Chapter 6: Logistic regression

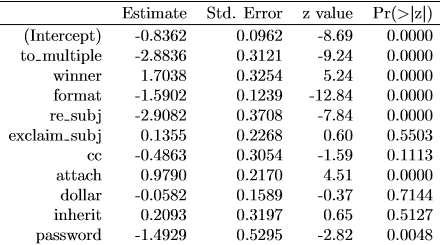
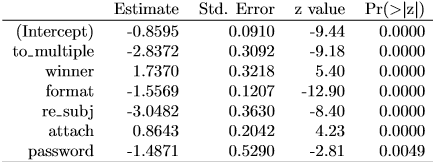
* **Logistic regression** = type of **generalized linear model (GLM)** for response variables where regular multiple regression does not work very well.
* Outcome variable in these settings often takes a form where residuals look completely diﬀerent from the normal distribution.
* GLMs can be thought of as a 2-stage modeling approach.
* **1) Model outcome using a probability distribution**, such as binomial or Poisson.
* **2) Model parameter of above distribution using a collection of predictors + a special form of multiple regression**.
* Ex: Email data set collected from a single email account 🡺 developing basic spam ﬁlter
* Outcome, **spam** = encoded to take value 0 when message = not spam + 1 when is spam.
* Task = build appropriate model that classiﬁes messages as spam or not spam using email characteristics coded as predictors.

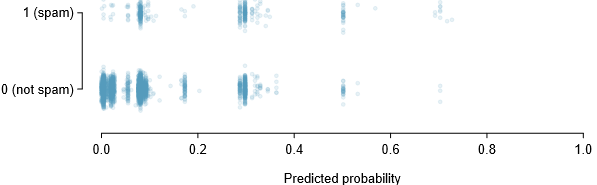


* Categorical + numerical predictors can be used in logistic regression.
* If outliers = present in predictors, corresponding observations may be especially inﬂuential on the resulting model.
* This = the motivation for omitting numerical variables, such as # of characters + line breaks in emails, which exhibited extreme skew.
* Could resolve this issue by transforming predictors (e.g. using a log-transform), but omitting this further investigation for brevity.
* Outcome for a GLM = denoted by Yi, where index **i** = used to represent observation I 🡺 represents whether email i is spam (Yi = 1) or not (Yi = 0).
* Predictors = represented as x(1,i) = value of predictor 1 for observation i, etc.
* Logistic regression = GLM where outcome = 2-level categorical variable.
* Outcome, Yi, takes value 1 (spam) w/ probability = **pi** (the one modeled in relation to the predictors) + value 0 w/ probability **(1 − pi)**.
* Logistic regression relates probability an email = spam to the predictors through a framework much like that of multiple regression: 
* Want to choose a **transformation** in that makes practical + mathematical sense.
* Ex: want a transformation that makes the range of possibilities on the LHS of the = the range of possibilities for the RHS
* If no possible transformation for this equation, the LHS could only take values between 0-1, but RHS could take values *outside* this range.
* Common transformation for pi = the **logit** transformation 🡺 

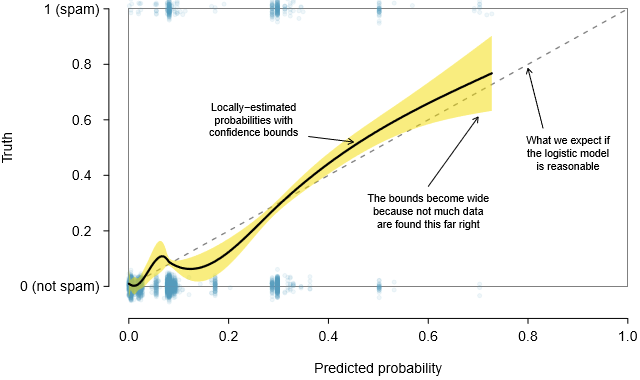




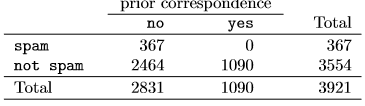
* For spam example: 10 predictors, so **k = 10**.
* Model isn’t very intuitive, but still has some resemblance to multiple regression, even if interpretation of the coeﬃcients is more complex.
* Example 6.20 Here we create a spam ﬁlter w/ a single predictor: **to\_multiple** = indicates whether more than 1 email address was listed in the “To” ﬁeld.
* Following model was ﬁt: 
* If randomly selected email has just 1 address in “To”, the probability it is spam = **-2.12 – 1.81\*0**
* Solving for pi:  = piˆ = **0.11 = 11%**
* If > 1 address listed, then model equation 🡺 −2.12−1.81×1 = −3.93, which corresponds to a probability **piˆ = 0.02**.
* Could examine -2.12 and -3.93 in plot to estimate probability before formally calculating value.
* To convert from values on the regression-scale (e.g. -2.12, -3.93), use following formula, which is the result of solving for pi in the regression model: 
* As w/ most applied data problems, we substitute **point estimates** for the **parameters** (β) so that we may make use of this formula.
* While info about whether email is addressed to multiple people = helpful start in classifying spam, probabilities of 11% + 2% = not dramatically diﬀerent, + neither provides very strong evidence about which particular messages = spam.
* To get more precise estimates, include more variables
* Full Model: Trimmed model: 
* 6.21 Examine summary of reduced model 🡪Is the **multiple row** point estimate the same as before, -1.81, or is it diﬀerent? Explain why this might be
* It’s different b/c it’s coefficient estimate is different (-2.837), due to interaction effects with new predictors that’re added to the model
* Point estimates change a little or a lot depending on which other predictors are included in model, usually due to collinearity in the predictors.
* 6.22 If incoming email has the word “winner” in it, will this raise or lower the model’s calculated probability that the incoming email is spam?
* It will increase it, as seen by the positive coefficient estimate
* 6.23 Suppose same email was in HTML **format**. Does this characteristic increase or decrease the probability that the email is spam according to the model?
* Format’s negative coefficient is not negative enough to cancel out the increase from **winner**
* 6.22 + 6.23 highlight a key feature of logistic + multiple regression 🡺 some email characteristics push classiﬁcation in direction of spam + others push it in opposite direction.
* If implementing a spam ﬁlter using model above, each future email analyzed would fall into 1 of 3 categories based on characteristics:
* 1. characteristics generally indicate not spam, so resulting spam probability = low, say, < 0.05
* 2. characteristics generally indicate spam, so resulting spam probability = large, say, > 0.95
* 3. characteristics roughly balance out in terms of evidence for + against spam + spam probability falls in remaining range, meaning email cannot be adequately classiﬁed as spam or not
* If managing an email service, must think about what should be done in each of these instances
* Email app 🡺 usually just 2 possibilities: ﬁlter email from inbox into “spambox”, or let email in inbox
* 6.24 The 1st + 2nd scenarios = intuitive. How should we handle emails in the third category?
* In this particular application, err on side of sending more mail to inbox rather than mistakenly putting good messages in the spambox (\*\*\***prefer false negatives to false positive\*\*\***)
* Emails in 1st + last categories = inbox, those in 2nd scenario = spambox.
* 6.25 Suppose we apply above logistic model + 100 messages are placed in spambox over 3 months. If we used the guidelines above for putting messages into the spambox, about how many legitimate (non-spam) messages would you expect to ﬁnd among the 100 messages?
* 95 🡺 proposed cutoﬀ for predicted probability of 0.95 for spam.
* Worst case scenario, all messages in spambox had the *minimum* probability = ~0.95.
* Thus, should expect to ﬁnd ~5 or fewer legitimate messages placed in the spambox.
* Spam ﬁlter guidelines above = okay to allow up to 5% of messages in spambox to be real messages.
* To make harder to classify messages as spam, use a cutoﬀ of 0.99
* 2 eﬀects
* raises standard for what can be classiﬁed as spam + reduces # of good emails classiﬁed as spam
* will also fail to correctly classify an increased fraction of spam messages
* No matter the complexity + conﬁdence in our model, **practical considerations** = absolutely crucial to making a helpful spam ﬁlter.
* Without them, could actually do more harm than good by using our statistical model.
* 2 key conditions for ﬁtting a logistic regression model:
* 1. Each predictor = linearly related to logit(pi) if all other predictors are held constant.
* 2. Each outcome Yi = independent of other outcomes.
* 1st condition of logistic regression model = not easily checked w/out a fairly sizable amount of data.
* We have 3,921 emails in our data set
* Visualize these data by plotting the true classiﬁcation of emails vs. model’s ﬁtted probabilities



* Vast majority of emails (spam or not) still have ﬁtted probabilities < 0.5
* Noise = added to each point so those w/ nearly identical values aren’t plotted exactly on top of one another = makes it possible to see more observations.
* May at ﬁrst seem discouraging to have no emails w/ ﬁtted probability > 0.75.
* Can improve model w/ “better” variables
* To assess quality of our model, might ask “if we look @ emails modeled as having a 10% chance of being spam, do we ﬁnd about 10% of them actually are spam?”
* Can borrow an advanced statistical method, **natural splines** = estimates **local probability** over the region 0.00-0.75 (largest predicted probability was 0.73, so we avoid extrapolating).
* All you need to know about **natural splines** = used to **ﬁt ﬂexible lines rather than straight lines.**



* solid black line = **empirical estimate** of the probability for observations based on their predicted probabilities (conﬁdence bounds also shown for this line) fit using natural splines
* small amount of noise was added to observations to allow more observations to be seen
* If the logistic model ﬁts well, curve should closely follow dashed **y = x** line.
* Shading represents conﬁdence bound for the curved line to clarify what ﬂuctuations might plausibly be due to chance.
* Even w/ conﬁdence bound: see weaknesses in the 1st model assumption 🡺 solid curve + its conﬁdence bound dips below dashed line from about 0.1-0.3 + drifts above it from about 0.35-0.55
* These deviations indicate model relating the parameter to the predictors does NOT closely resemble true relationship.
* Could evaluate the 2nd logistic regression model assumption (independence of outcomes) using model residuals, calculated same way as w/ multiple regression: observed - expected outcome.
* **For logistic regression, expected value of the outcome = ﬁtted probability for the observation**, + the residual written as
* Could plot these residuals against a variety of variables or in order of collection
* However, since model will need to be revised to eﬀectively classify spam + we’ve already seen similar residual plots above earlier, we won’t investigate the residuals here.
* If building spam ﬁlter for email service that managed many accounts (e.g. Gmail), would spend much more time thinking about additional variables that could be useful in classifying spam + also would use transformations or other techniques to help include strongly skewed numerical predictors
* List of variables we think might be useful:
* (1) Indicator to represent whether there was prior 2-way correspondence w/ message’s sender.
* (2) Indicator = 1 if sender of message has previously sent messages ﬂagged as spam.
* (3) Indicator that ﬂag emails that contain links included in previous spam messages.
* The variables described above take 1 of 2 approaches.
* Variable (1) = specially designed to capitalize on fact spam is rarely sent between individuals that have 2-way communication.
* (2) + (3) = specially designed to ﬂag common spammers or spam messages.
* Would have to verify each variable is eﬀective using the data, but these seem like promising ideas.
* See contingency table for spam + for new variable (1) above



* For the 1,090 emails w/ correspondence in the preceding 30 days, not 1 was spam 🡺 suggests variable (1) would be very eﬀective at accurately classifying some messages as not spam.
* W/ this single variable, we’d be able to send about 28% of messages through to inbox w/ conﬁdence almost none = spam.
* (2) and (3) would provide an excellent foundation for distinguishing messages coming from known spammers or messages that take a known form of spam.
* To utilize these, would need to build databases: 1 holding addresses of known spammers + 1 holding URLs found in known spam messages.
* Our access to such info is limited, so we cannot implement these
* However, if hired by an email service to build a spam ﬁlter, these would be important next steps.
* In addition to ﬁnding more + better predictors, we’d need to create a customized logistic regression model for each email account.
* May sound intimidating, but its complexity is not as daunting as it may at ﬁrst seem.
* Simple email variables, such as format, inclusion of certain words, + other circumstantial characteristics, provide helpful info for spam classiﬁcation.
* Many challenges remain, from better understanding logistic regression to carrying out necessary programming