Week 5 assignment: Binomial models

Will Stutz

15 February, 2016

About one out of eight women in the U.S. will develop breast cancer at some point in her lifetime. Early diagnoses help with treatment of this potentially fatal disease, and these diagnoses can be made based on a variety of cytological metrics evaluated via biopsy. Your job today is to develop a model that classifies tumors as malignant or benign based on these metrics. The student(s) with the most predictive model will get a prize.

The data are in the `breast_cancer.csv` file. Details for this dataset can be found on the UCI machine learning data repository, which is useful if you ever need data to play with. I split the data into two groups at random: the *training* data, which you'll use to estimate parameters, and the *test* data, which we'll use to evaluate the predictive power of the model. There is a column in the data called `group`, which indicates whether an observation is part of the training or test set.

Data exploration

As usual, you will want to explore the data before constructing any statistcal models. Only explore the training data, and do not use the test data for data exploration/visualization. We will pretend that we don't have access to the test data yet.

```
# some useful libraries
library(ggplot2)
library(rstan)
library(reshape)
library(tidyr)

# set some options
options(mc.cores = parallel::detectCores())

# upload the data
dat <- read.csv("data/breast_cancer.csv")

# let's go ahead and just pull the training data output
train <- dat[dat$group == "train",] %>% droplevels

# how big is the data set?
dim(train)
```

```
## [1] 333  13
```

```
  # 333 observations and 13 data columns

# what kind of data do we have?
head(train)
```

```
##       id clump_thickness size_uniformity shape_uniformity
## 1  95719               6              10               10
## 2 128059               1               1                1
```

```
## 3 145447               8               4               4
## 4 167528               4               1               1
## 5 183913               1               2               2
## 6 242970               5               7               7
##   marginal_adhesion epithelial_size bare_nuclei bland_chromatin
## 1                10               8         10               7
## 2                 1               2          5               5
## 3                 1               2          9               3
## 4                 1               2          1               3
## 5                 1               2          1               1
## 6                 1               5          8               3
##   normal_nucleoli mitoses   cohort group malignant
## 1              10       7 cohort_1 train         1
## 2               1       1 cohort_1 train         0
## 3               3       1 cohort_1 train         1
## 4               6       1 cohort_1 train         0
## 5               1       1 cohort_1 train         0
## 6               4       1 cohort_1 train         0
```

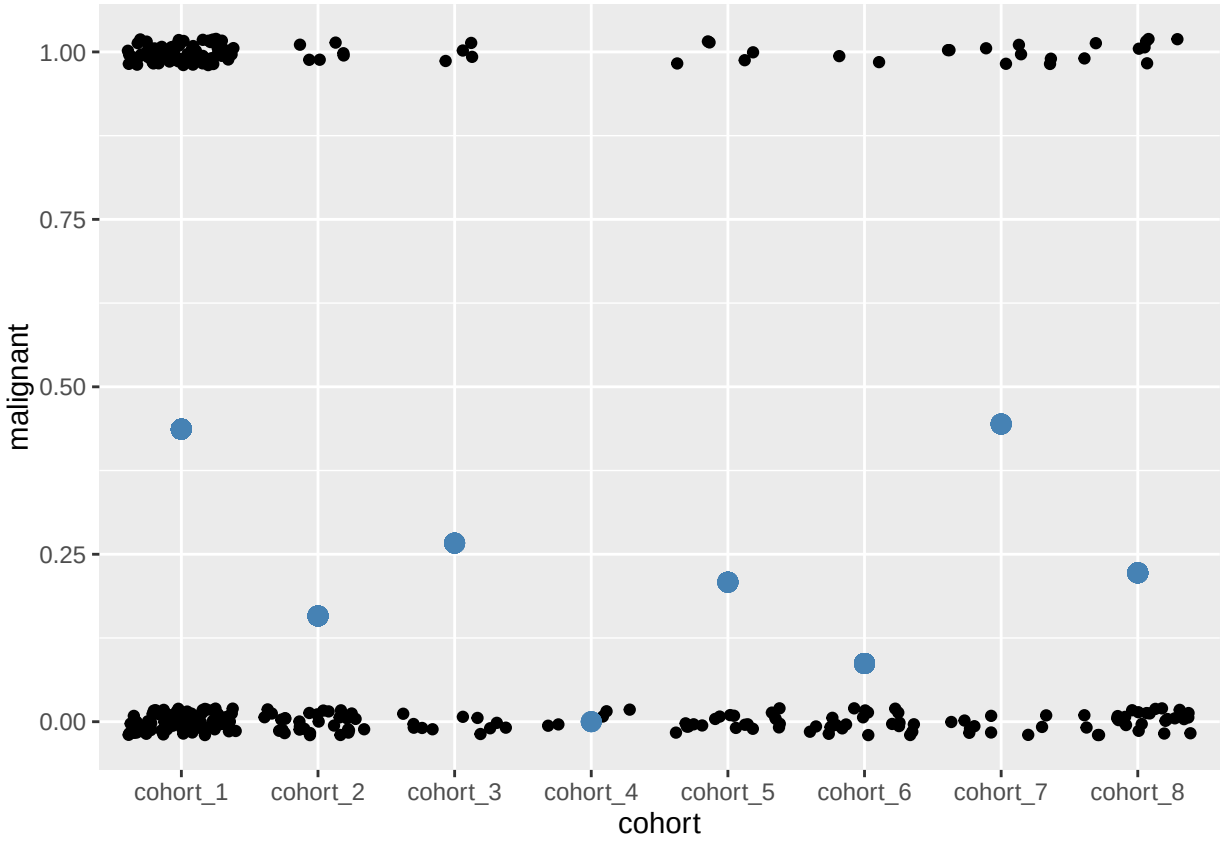
```
summary(train)
```

```
##        id          clump_thickness  size_uniformity  shape_uniformity
##  Min.   :  95719   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 896404   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median :1171845   Median : 4.000   Median : 1.000   Median : 1.000
##  Mean   :1073427   Mean   : 4.333   Mean   : 3.165   Mean   : 3.189
##  3rd Qu.:1238633   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000
##  Max.   :8233704   Max.   :10.000   Max.   :10.000   Max.   :10.000
##
##  marginal_adhesion epithelial_size   bare_nuclei     bland_chromatin
##  Min.   : 1.000    Min.   : 1.000   Min.   : 0.000   Min.   : 1.000
##  1st Qu.: 1.000    1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 2.000
##  Median : 1.000    Median : 2.000   Median : 1.000   Median : 3.000
##  Mean   : 2.796    Mean   : 3.165   Mean   : 3.351   Mean   : 3.426
##  3rd Qu.: 3.000    3rd Qu.: 4.000   3rd Qu.: 5.000   3rd Qu.: 4.000
##  Max.   :10.000    Max.   :10.000   Max.   :10.000   Max.   :10.000
##
##  normal_nucleoli    mitoses            cohort      group
##  Min.   : 1.00   Min.   : 1.000   cohort_1:174   train:333
##  1st Qu.: 1.00   1st Qu.: 1.000   cohort_2: 38
##  Median : 1.00   Median : 1.000   cohort_8: 36
##  Mean   : 2.82   Mean   : 1.634   cohort_5: 24
##  3rd Qu.: 3.00   3rd Qu.: 1.000   cohort_6: 23
##  Max.   :10.00   Max.   :10.000   cohort_7: 18
##                                   (Other) : 20
##    malignant
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3273
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
```

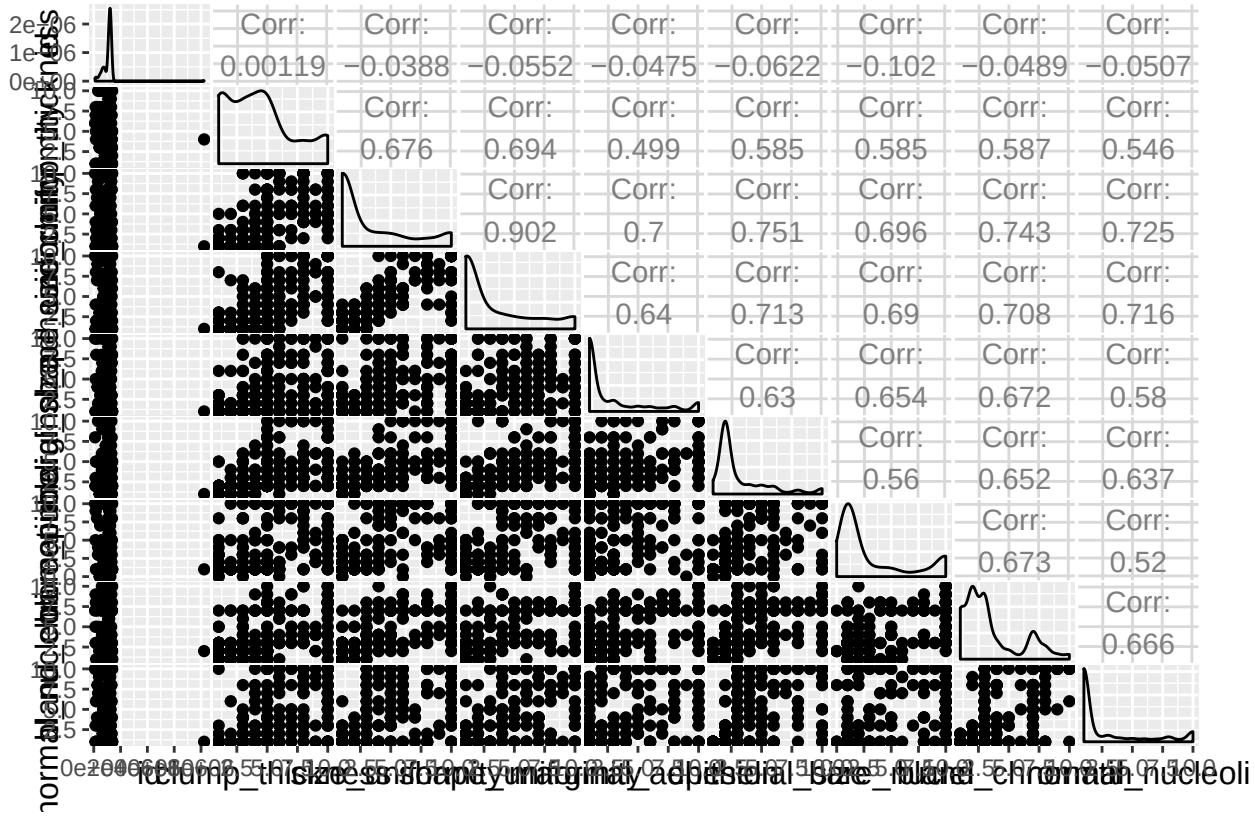
```
  # looks like a bunch of integer variables that vary between 1 and 10
  # 'malignant' must be whether the tumor was malignant or not

# how many cohorts?
table(train$cohort)
```

```
##
## cohort_1 cohort_2 cohort_3 cohort_4 cohort_5 cohort_6 cohort_7 cohort_8
##      174       38       15        5       24       23       18       36
```

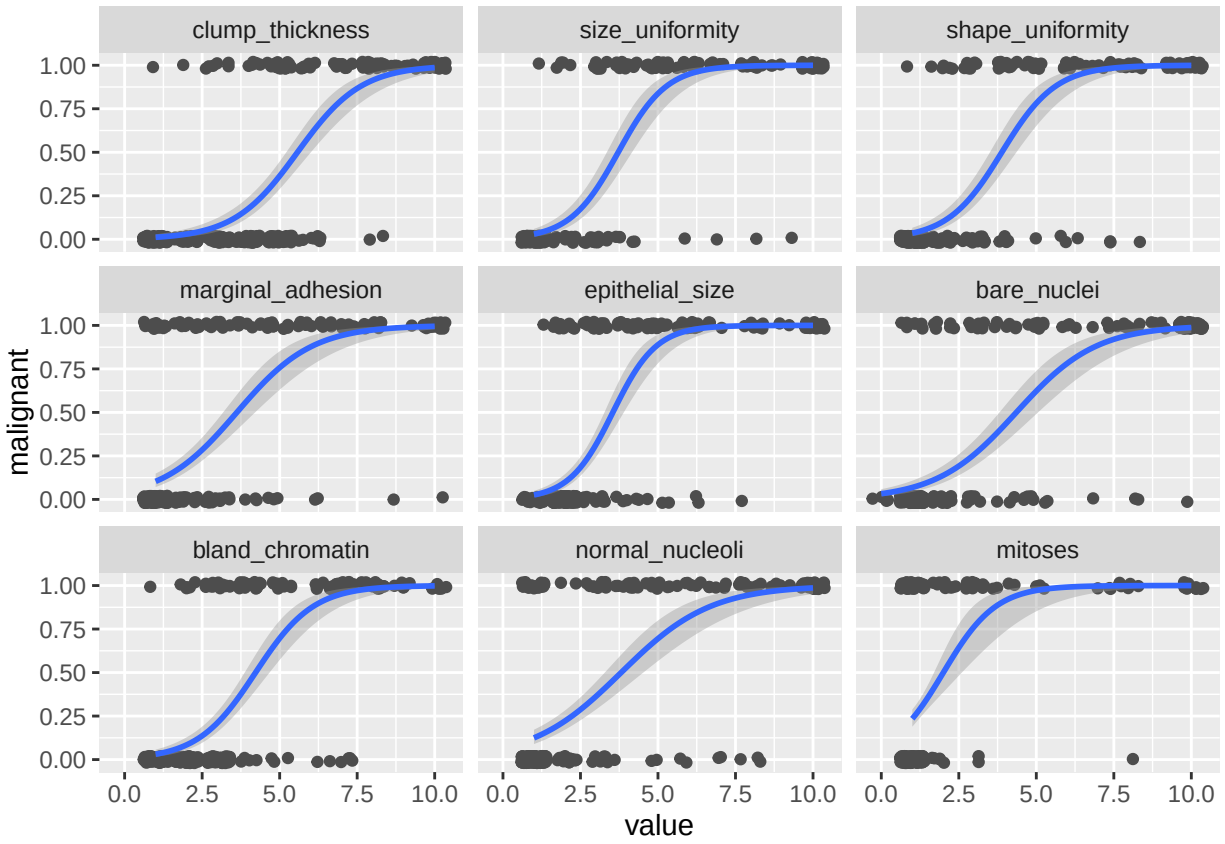
```
  # 8!

# are there differences between cohorts?

  # calculate means
  cohort_means <- data.frame(cohort = levels(train$cohort),
    prob = as.vector(tapply(train$malignant, train$cohort, mean)))

  # add to Data
  train <- merge(train,cohort_means,by = c("cohort"))

  # plot raw data with cohort means
  ggplot(data = train, aes(cohort,malignant)) +
    geom_jitter(height = 0.05) +
    geom_point(aes(y = prob), color = "steelblue", size = 3)
```



```
# let's melt all the predictor variables into a single column
train_melt <- melt(train, id.vars = c("id","cohort","group","malignant","prob"))

# are there correlations between the predictors?
library(GGally) # for the ggpairs function
ggpairs(train[,c(2:10)])
```




```
  # hard to see but shape and size uniformity are highly correlated

# plot the predictor data versus malignancy
ggplot(data = train_melt, aes(value, malignant)) +
  facet_wrap(~variable) +
  geom_jitter(height = 0.05, color = "grey30") +
  geom_smooth(method = "glm", method.args = list(family = "binomial"))
```




```
  # all are positively associated with the probability of malignancy
```

Model structure

What is your model? Write it out in LaTeX. Hint: you will want to use a design matrix.

$y \sim Bernoulli(p)$

$logit(p) = X\beta$

What is your Stan model statement?

```
data {
  // integer inputs
  int n;  // the number of samples
  int n_pred;  // the number of predictors
  int n_cohort;  // the number of cohorts

  // integer vector inputs
  int<lower=0, upper=1> y[n];   // observed malignancies

  // design matrix
  matrix[n,n_pred + n_cohort] X;
}

parameters {
```

```
  // vector intercpet, betas for predictors and cohort means
  vector [n_pred + n_cohort] beta; //
}

model {

  // define priors for continuous predictors
  beta[1:n_pred] ~ cauchy(0, 3);

  // define priors for cohort effects
  beta[n_pred + 1:11] ~ cauchy(0,5);
  beta[12] ~ normal(0,5);
  beta[13:n_pred+n_cohort] ~ cauchy(0,5);

  // define the likelihood
  y ~ bernoulli_logit(X*beta);

}
```

Building and understanding the design matrix

We mentioned that you would want to use a design matrix. Specifically, your model should be of the form:

$y \sim Bernoulli(p)$

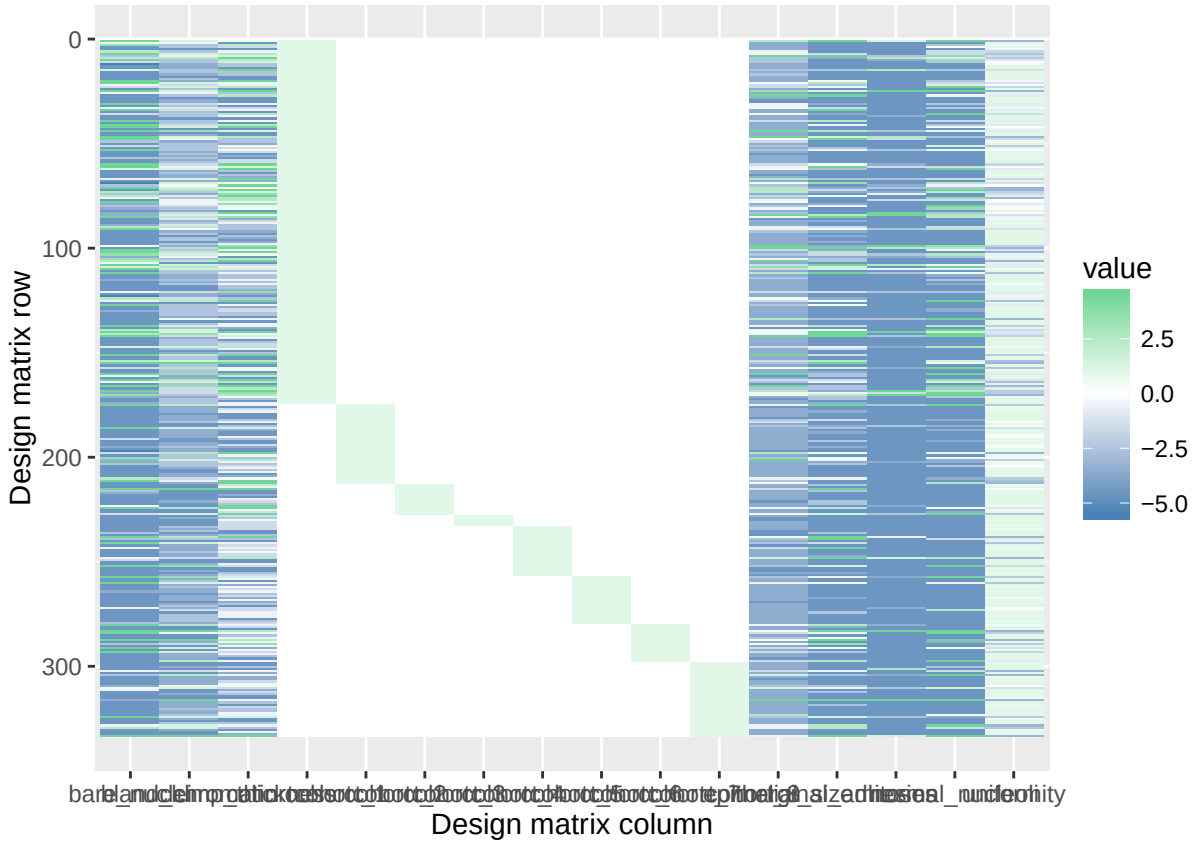
And the probability of malignancy p is modeled using a logit-link:

$log\Big(\frac{p}{1-p}\Big) = X\beta$

The design matrix X contains the tumor features, and also dictates the interpretation of the coefficients β. In the code block below, construct your design matrix, creating an object called `X`. The included code will make an image plot of your design matrix with a horrendous color scheme. Once you fill in your code, set the argument `eval = TRUE` inside of the curly braces at the beginning of the code chuck (this is a chunk option), otherwise the code chunk will not be evaluated when you're knitting your pdf.

```
# calculate principal components of uniformity
unif_pca <- prcomp(~shape_uniformity + size_uniformity, data = train,
    center = TRUE, scale = TRUE)

# add to data
train$uniformity <- unif_pca$x[,1]

# center variables
train_centered <- train # create new data frame
train_centered[,c(3,6:11)] <- train_centered[,c(3,6:11)] - 5.5
  # center variables at 5.5 since they are on a 1-10 scale


# define your design matrix below
X <- model.matrix(~0 + clump_thickness + uniformity + marginal_adhesion +
  epithelial_size + bare_nuclei + bland_chromatin + normal_nucleoli + mitoses
  + cohort, data = train_centered)


# the code below will plot your design matrix
```

```
mX <- melt(X)
ggplot(mX, aes(x = X2, y = X1)) +
  geom_raster(aes(fill = value)) +
  scale_y_reverse() +
  xlab('Design matrix column') +
  ylab('Design matrix row') +
  scale_fill_gradient2(low = "steelblue",  mid = "white", high = "seagreen3")
```



For each column of X you will get a coefficient, one element in β. For instance, the coefficient β_1 will be associated with the first column in X, which we might denote $X[,1]$, to borrow some R syntax. There's no sense in estimating parameters if you don't know what they mean (Abraham Lincoln said that), so below, list each element in β and briefly describe what it represents/how you would interpret it:

1. β_1 represents *the increase in the logit probability that a tumor is malignant given an increase of 1 point in clump thickness*
2. β_2 represents *the increase in the logit probability that a tumor is malignant given an increase of 1 point in uniformity*
3. β_3 represents *the increase in the logit probability that a tumor is malignant given an increase of 1 point in marginal adhesion*
4. β_4 represents *the increase in the logit probability that a tumor is malignant given an increase of 1 point in epithelial size*
5. β_5 represents *the increase in the logit probability that a tumor is malignant given an increase of 1 point in bare_nuclei*

6. β_6 represents *the increase in the logit probability that a tumor is malignant given an increase of 1 point in bland chromatin*
7. β_7 represents *the increase in the logit probability that a tumor is malignant given an increase of 1 point in normal nucleoli*
8. β_8 represents *the increase in the logit probability that a tumor is malignant given an increase of 1 point in mitoses*
9. β_9 represents *the logit probability that a tumor is malignant it the carrier was in cohort 1*
10. β_{10} represents *the logit probability that a tumor is malignant it the carrier was in cohort 2*
11. β_{11} represents *the logit probability that a tumor is malignant it the carrier was in cohort 3*
12. β_{12} represents *the logit probability that a tumor is malignant it the carrier was in cohort 4*
13. β_{13} represents *the logit probability that a tumor is malignant it the carrier was in cohort 5*
14. β_{14} represents *the logit probability that a tumor is malignant it the carrier was in cohort 6*
15. β_{15} represents *the logit probability that a tumor is malignant it the carrier was in cohort 7*
16. β_{16} represents *the logit probability that a tumor is malignant it the carrier was in cohort 8*

Parameter estimation

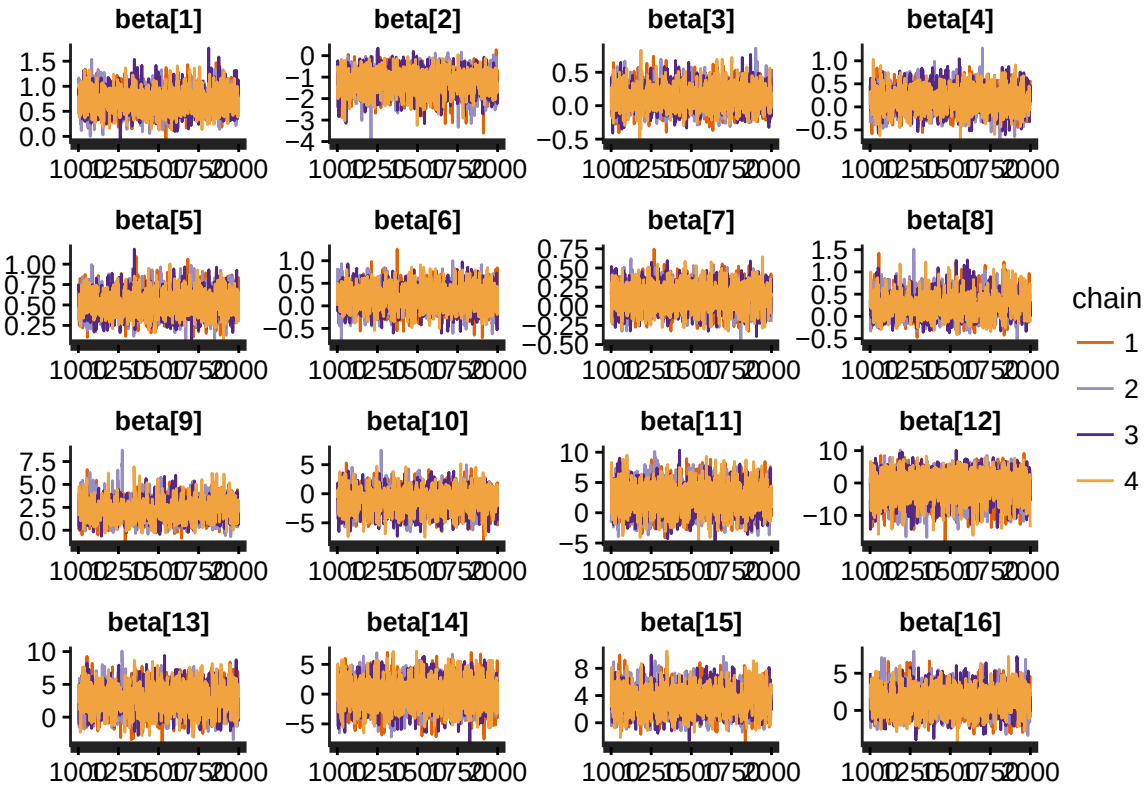
Use the **training** data to estimate your model's parameters (`group == 'train'`). Do not use the **test** data yet. Make sure that the MCMC algorithm has converged before moving forward.

```
# build the Data
stan_d <- list(n = nrow(train),
  n_pred = 8,
  n_cohort = length(levels(train$cohort)),
  X = X,
  y = train$malignant)

# fit Model
tumor_fit <- stan("tumor_glm_will.stan", data = stan_d)

# check Rhat
print(tumor_fit)
```

```
## Inference for Stan model: tumor_glm_will.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## beta[1]    0.71    0.01 0.23   0.30   0.56   0.71   0.86   1.18  1752    1
## beta[2]   -1.23    0.01 0.54  -2.36  -1.57  -1.20  -0.85  -0.25  1994    1
## beta[3]    0.13    0.00 0.17  -0.20   0.01   0.12   0.24   0.48  2454    1
## beta[4]    0.16    0.00 0.25  -0.32   0.00   0.16   0.33   0.67  2672    1
## beta[5]    0.52    0.00 0.14   0.25   0.43   0.51   0.61   0.83  1722    1
## beta[6]    0.19    0.00 0.25  -0.30   0.02   0.19   0.36   0.69  2866    1
## beta[7]    0.12    0.00 0.16  -0.19   0.01   0.12   0.23   0.42  2675    1
## beta[8]    0.26    0.01 0.26  -0.19   0.08   0.24   0.42   0.84  1104    1
```

```
## beta[9]    2.21    0.03 1.14   0.21   1.42   2.12   2.89   4.71  1057    1
## beta[10]  -1.16    0.05 1.93  -4.94  -2.48  -1.18   0.14   2.66  1602    1
## beta[11]   2.38    0.05 2.23  -1.95   0.87   2.37   3.81   6.91  2446    1
## beta[12]  -1.23    0.08 4.30 -10.37  -4.12  -0.88   1.89   5.95  3093    1
## beta[13]   2.76    0.05 1.95  -1.10   1.48   2.75   4.05   6.60  1731    1
## beta[14]  -0.01    0.05 2.40  -4.81  -1.68   0.00   1.63   4.61  2691    1
## beta[15]   3.35    0.05 1.86  -0.16   2.06   3.28   4.61   7.11  1607    1
## beta[16]   1.46    0.05 1.64  -1.69   0.36   1.45   2.53   4.73  1313    1
## lp__     -40.09    0.09 3.11 -47.18 -41.91 -39.68 -37.90 -35.13  1206    1
##
## Samples were drawn using NUTS(diag_e) at Mon Feb 15 16:28:27 2016.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
# check traceplots
rstan::traceplot(tumor_fit, pars = "beta")
```



Out of sample predictive power

One measure of a model's ability to predict new data is the log likelihood of new data, given the parameters of the model $[\tilde{y} \mid \theta]$, where $\tilde{y}$ is the new data (the **test** or **validation** data), and the parameters θ have been estimated from other data (e.g., the **training** data).

Hints:

- this is done most easily via a new design matrix X_{test}, which can be multiplied by the vector of model parameters, and must be declared in the `data` block

- make sure that if you used any feature scaling or centering in the training data, that the exact same scaling/centering schemes are applied to the test set
- you'll use the `generated quantities` block to calculate the log-likelihood of the test data
- you can obtain the joint log likelihood with the `bernoulli_logit_log` function in Stan, and I wrote a generated quantities model block for you below, which should be the last block in your new Stan model statement

What is your updated Stan model?

```
data {
  // integer inputs
  int n;  // the number of samples
  int n_pred;  // the number of predictors
  int n_cohort;  // the number of cohorts
  int n_test; // number of individuals in the test data


  // integer vector inputs
  int<lower=0, upper=1> y[n];   // observed malignancies
  int<lower=0, upper=1> y_test[n_test];   // observed for test data

  // design matrix
  matrix[n,n_pred + n_cohort] X;
  matrix[n_test, n_pred + n_cohort] X_test;
}

parameters {

  // vector intercept, betas for predictors and cohort means
  vector [n_pred + n_cohort] beta; //
}

model {


  // define priors for continuous predictors
  beta[1:n_pred] ~ cauchy(0, 3);

  // define priors for cohort effects
  beta[n_pred + 1:11] ~ cauchy(0,5);
  beta[12] ~ normal(0,5); #shrinks estimate for cohort 4 towards 50%
  beta[13:n_pred+n_cohort] ~ cauchy(0,5);

  // define the likelihood
  y ~ bernoulli_logit(X*beta);

}

generated quantities {
  real loglik_test;
  vector[n_test] logit_p_test;

  logit_p_test <- X_test * beta;
  loglik_test <- bernoulli_logit_log(y_test, logit_p_test);
```

```
  //returns the sum of the log likelihoods (the joint log-likelihood)
}
```

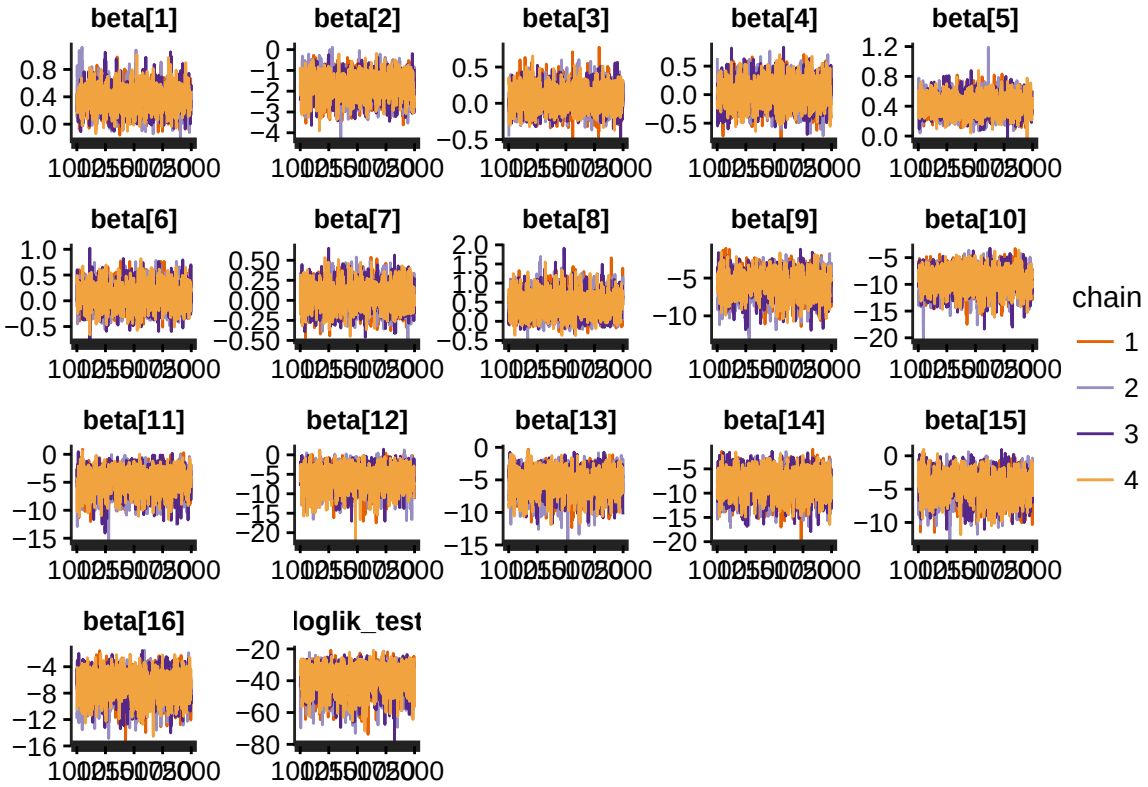
Acquire the posterior distribution of the model parameters and the holdout log likelihood.

```
# calculate new PCA using all the data
dat$uniformity <- prcomp(~size_uniformity + shape_uniformity, data = dat,
  center = TRUE, scale = TRUE)$x[,1]

# center variables
dat_centered <- dat # create new data frame
dat_centered[,c(2,5:10)] <- dat_centered[,c(2,5:10)] - 5.5
  # center variables at 5.5 since they are on a 1-10 scale

# pull test data
test_centered <- dat[dat$group == "test",]

# pull training data
train_centered <- dat[dat$group == "train",]

# define training design matrix
X <- model.matrix(~0 + clump_thickness + uniformity + marginal_adhesion +
  epithelial_size + bare_nuclei + bland_chromatin + normal_nucleoli + mitoses
  + cohort, data = train_centered)

# define test design matrix
X_test <- model.matrix(~0 + clump_thickness + uniformity + marginal_adhesion +
  epithelial_size + bare_nuclei + bland_chromatin + normal_nucleoli + mitoses
  + cohort, data = test_centered)

# build the Data
stan_d <- list(n = nrow(train_centered),
  n_pred = 8,
  n_cohort = length(levels(train_centered$cohort)),
  n_test = nrow(test_centered),
  X = X,
  x_test = X_test,
  y = train_centered$malignant,
  y_test = test_centered$malignant)

# fit Model
tumor_fit_test <- stan("tumor_glm_test_will.stan", data = stan_d)

# check Rhat
print(tumor_fit_test, pars = c("beta","loglik_test"))
```

```
## Inference for Stan model: tumor_glm_test_will.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##               mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff
## beta[1]       0.37    0.00 0.18   0.04   0.25   0.36   0.49   0.72  1487
## beta[2]      -1.77    0.02 0.57  -2.97  -2.15  -1.76  -1.38  -0.73  1424
```

```
## beta[3]       0.08    0.00 0.16  -0.23  -0.03   0.07   0.18   0.40  2637
## beta[4]       0.02    0.00 0.23  -0.42  -0.13   0.02   0.17   0.48  2262
## beta[5]       0.42    0.00 0.13   0.18   0.33   0.42   0.51   0.68  1687
## beta[6]       0.09    0.00 0.23  -0.35  -0.06   0.09   0.24   0.54  2297
## beta[7]       0.05    0.00 0.15  -0.25  -0.05   0.05   0.15   0.36  2246
## beta[8]       0.44    0.01 0.30  -0.06   0.23   0.42   0.63   1.09  2136
## beta[9]      -5.87    0.06 1.66  -9.38  -6.92  -5.77  -4.73  -2.87   844
## beta[10]     -9.23    0.06 2.11 -13.70 -10.53  -9.11  -7.78  -5.45  1116
## beta[11]     -4.94    0.06 2.11  -9.55  -6.27  -4.79  -3.39  -1.35  1335
## beta[12]     -6.00    0.07 2.91 -12.65  -7.72  -5.69  -3.94  -1.20  1754
## beta[13]     -5.49    0.05 1.89  -9.67  -6.65  -5.34  -4.14  -2.24  1206
## beta[14]     -7.95    0.08 2.69 -13.79  -9.61  -7.69  -6.02  -3.33  1189
## beta[15]     -4.50    0.06 1.94  -8.64  -5.76  -4.37  -3.16  -1.04  1177
## beta[16]     -7.01    0.06 2.02 -11.28  -8.34  -6.85  -5.62  -3.42  1143
## loglik_test -38.90    0.15 7.83 -57.25 -43.35 -37.87 -33.25 -26.60  2585
##             Rhat
## beta[1]        1
## beta[2]        1
## beta[3]        1
## beta[4]        1
## beta[5]        1
## beta[6]        1
## beta[7]        1
## beta[8]        1
## beta[9]        1
## beta[10]       1
## beta[11]       1
## beta[12]       1
## beta[13]       1
## beta[14]       1
## beta[15]       1
## beta[16]       1
## loglik_test    1
##
## Samples were drawn using NUTS(diag_e) at Mon Feb 15 16:28:46 2016.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

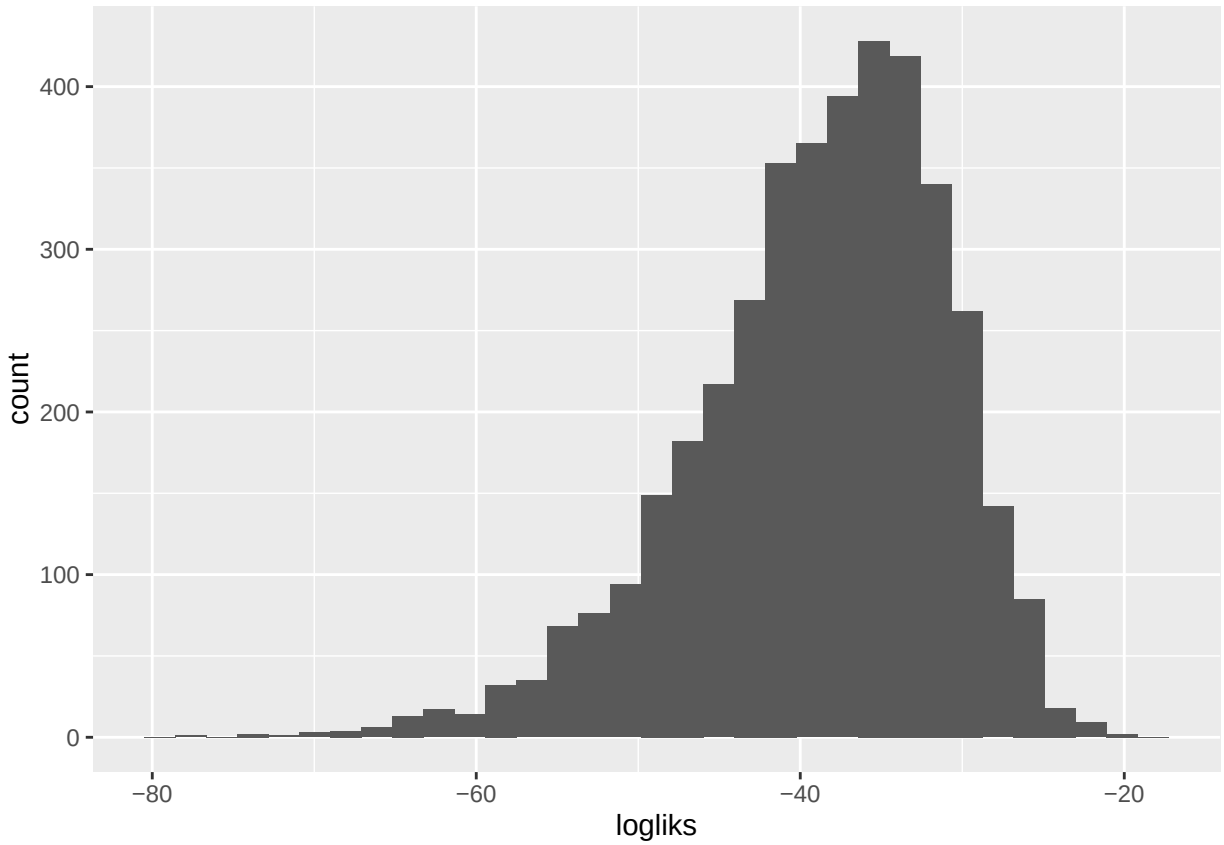
```
# check traceplots
rstan::traceplot(tumor_fit_test, pars = c("beta","loglik_test"))
```



Make a histogram of the holdout log likelihood and report the posterior mean along with a 95% credible interval.

```
# extract summed log-likelihoods for each draw
logliks <- rstan::extract(tumor_fit_test)$loglik_test

# create histogram
qplot(logliks, geom = "histogram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# calculate mean and 95% CI
data.frame(mean = mean(logliks),
  low_ci = quantile(logliks, prob = c(0.025)),
  hi_ci = quantile(logliks, prob = c(0.975)))
```

```
##           mean    low_ci     hi_ci
## 2.5% -38.90106 -57.24666 -26.60052
```

Showing predictions

The whole point of building this model is to predict whether a tumor is malignant based on some features. Plot the posterior probability of tumor malignance for each holdout tumor, and show the true tumor status in the same graph. Multiple graph types are possible here, but we do not recommend simply copying and pasting code from another example (so far about a quarter of plots made in this way have made sense). Instead, think hard about what sort of data display would be effective, and make that plot!

```
library(coda)
# extract logit probabilities for each test tumor
logit_probs <- as.mcmc(rstan::extract(tumor_fit_test)$logit_p_test)

# calculate mean and 95% credible intervals for each patient
prob_summary <- summary(logit_probs)

# create data frame for plotting
plot_data <- data.frame(patient = as.character(test_centered$id),
```

```
  cohort = test_centered$cohort,
  malignant = as.factor(test_centered$malignant),
  mean = prob_summary$statistics[,"Mean"],
  lwr = prob_summary$quantiles[,"2.5%"],
  upr = prob_summary$quantiles[,"97.5%"])

# plot
ggplot(data = plot_data, aes(patient, mean)) +
  geom_hline(yintercept = 0, lty = "dotted") +
  geom_errorbar(aes(ymin = lwr, ymax = upr), color = "grey70",width = 0.02) +
  geom_point(aes(color = malignant)) +
  facet_wrap(~cohort,ncol = 3,scales = "free_x") +
  labs(y = "logit probability of malignancy", x = "patient") +
  theme_bw() +
  theme(axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    panel.grid = element_blank())
```