Homework 3 - Chapter Question 2

Nickhil Sethi

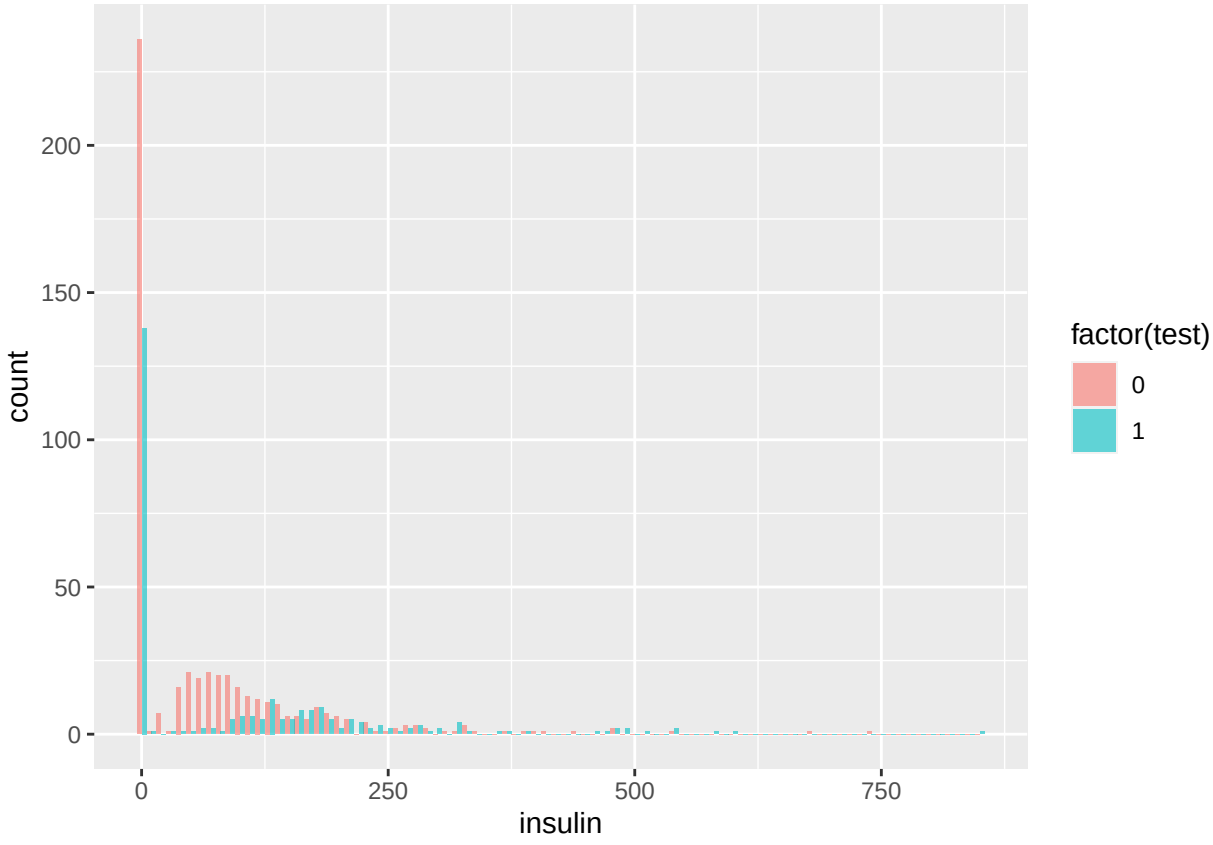
Load pima dataset:

```
data("pima")
```

2.a

See below for the interleaved histogram plot. One thing that does seem unbelievable about this plot – is the very large number of zeros on the insulin variable, both for positive and negative tests. This suggests to me either a possible coding error or 0 representing missing data.

```
ggplot(pima, aes(x=insulin, fill=factor(test))) +
geom_histogram(
  alpha=.6, position="dodge", binwidth=10)
```

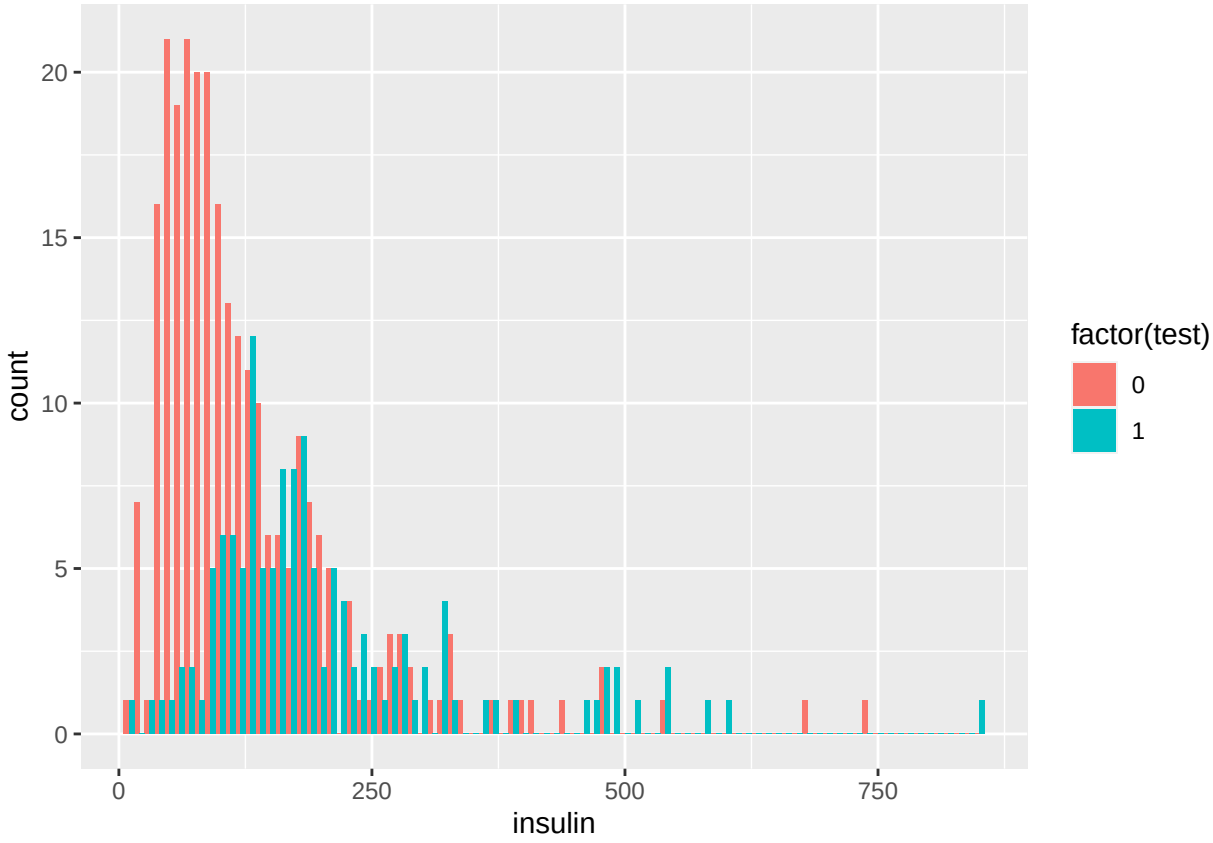


2.b

Without the incredible zero values, insulin appears to have a poisson distribution with mean conditional on the test outcome:

```
pima$insulin[pima$insulin == 0] <- NA
ggplot(pima, aes(x=insulin, fill=factor(test))) + geom_histogram(position="dodge", binwidth=10)
```

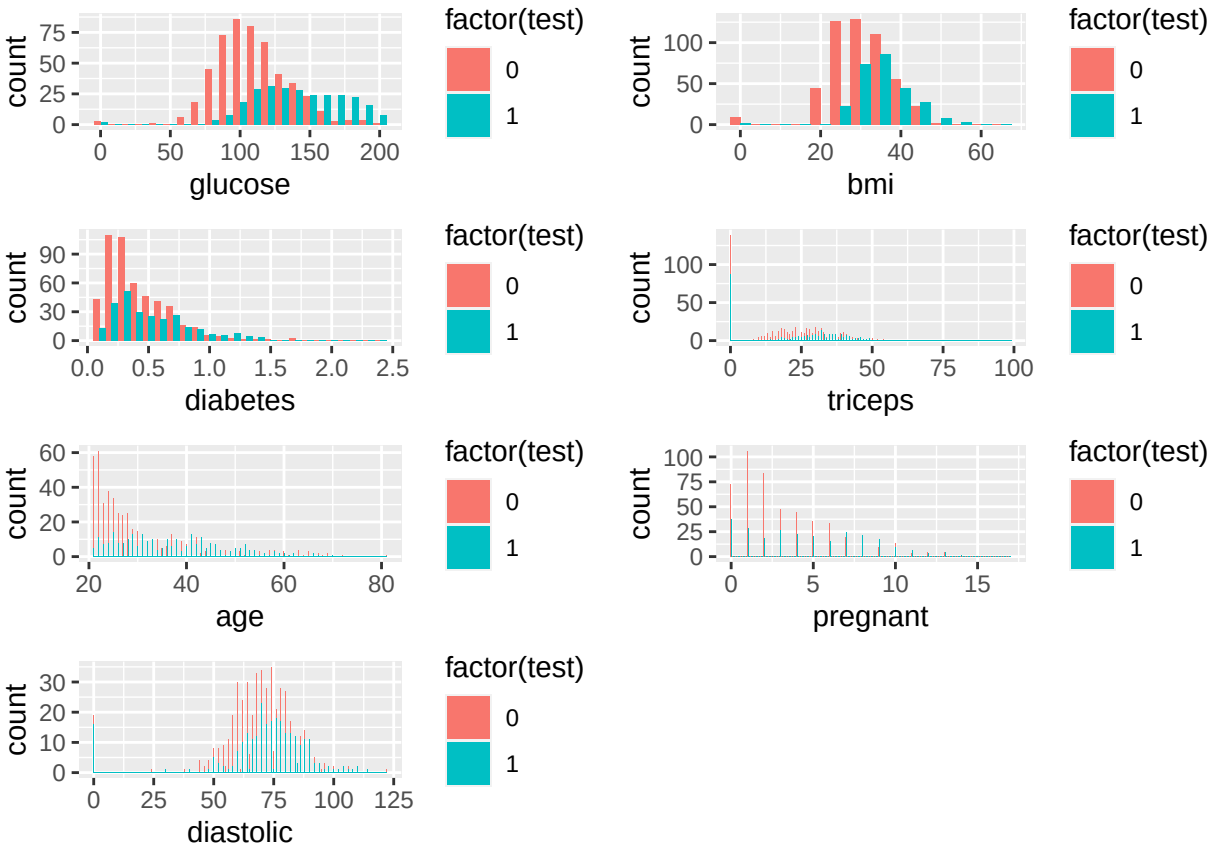
```
## Warning: Removed 374 rows containing non-finite values (stat_bin).
```



2.c

Let's plot the other variables and inspect for any suspect zero values:

```
g1 <- ggplot(pima, aes(x=glucose, fill=factor(test))) + geom_histogram(position="dodge", binwidth=10)

g2 <- ggplot(pima, aes(x=bmi, fill=factor(test))) + geom_histogram(position="dodge", binwidth=5)

g3 <- ggplot(pima, aes(x=diabetes, fill=factor(test))) + geom_histogram(position="dodge", binwidth=.1)

g4 <- ggplot(pima, aes(x=triceps, fill=factor(test))) + geom_histogram(position="dodge", binwidth=.1)

g5 <- ggplot(pima, aes(x=age, fill=factor(test))) + geom_histogram(position="dodge", binwidth=.1)

g6 <- ggplot(pima, aes(x=pregnant, fill=factor(test))) + geom_histogram(position="dodge", binwidth=.1)

g7 <- ggplot(pima, aes(x=diastolic, fill=factor(test))) + geom_histogram(position="dodge", binwidth=.1)

grid.arrange(g1, g2, g3, g4, g5, g6, g7, ncol=2)
```



It seems the following variables have incredible zeros:

```
pima$glucose[pima$glucose == 0] <- NA
pima$bmi[pima$bmi == 0] <- NA
pima$triceps[pima$triceps == 0] <- NA
pima$diastolic[pima$diastolic == 0] <- NA
nrow(pima)
```

```
## [1] 768
```

Now let's regress test on the other variables. 392 of the 768 rows in *pima* were used, with the rest being dropped due a column having a value of `NA`.

```
# number of rows
# in the pima dataset
nrow(pima)
```

```
## [1] 768
```

```
# fit the model
lmod <- glm(test ~ ., family=binomial, pima)
sumary(lmod)
```

```
##                Estimate  Std. Error z value  Pr(>|z|)
## (Intercept) -1.0041e+01  1.2177e+00 -8.2458 < 2.2e-16
## pregnant     8.2159e-02  5.5426e-02  1.4823  0.138250
## glucose      3.8270e-02  5.7677e-03  6.6351 3.242e-11
## diastolic   -1.4203e-03  1.1833e-02 -0.1200  0.904464
## triceps      1.1221e-02  1.7084e-02  0.6568  0.511279
## insulin     -8.2531e-04  1.3064e-03 -0.6317  0.527565
```

```
## bmi          7.0538e-02  2.7342e-02  2.5798  0.009885
## diabetes     1.1409e+00  4.2743e-01  2.6692  0.007603
## age          3.3952e-02  1.8382e-02  1.8470  0.064743
##
## n = 392 p = 9
## Deviance = 344.02123 Null Deviance = 498.09781 (Difference = 154.07657)
```

2.d

Running this model without *insulin* and *triceps* predictors, we see that 724 observations are now used in the model, indicating most of the NA values were in those two columns.

```
lmod2 <- glm(test ~ pregnant + glucose + diastolic
          + bmi + diabetes + age, family=binomial, pima)
sumary(lmod2)
```

```
##               Estimate Std. Error  z value  Pr(>|z|)
## (Intercept) -8.9621457  0.8208917 -10.9176 < 2.2e-16
## pregnant     0.1178628  0.0334181   3.5269 0.0004204
## glucose      0.0351944  0.0036049   9.7630 < 2.2e-16
## diastolic   -0.0089162  0.0086176  -1.0346 0.3008356
## bmi          0.0909264  0.0157398   5.7768 7.612e-09
## diabetes     0.9605147  0.3064152   3.1347 0.0017204
## age          0.0169443  0.0098344   1.7230 0.0848936
##
## n = 724 p = 7
## Deviance = 672.86106 Null Deviance = 931.93733 (Difference = 259.07627)
```

Now to test the significance of the smaller model against the larger model, we look at the difference in the deviances of the two models, which should be chi-squared distributed under null. As we can see below, the deviance of the smaller model is much higher than larger model, indicating a better fit; in fact the p-value for this difference is below machine precision, so it prints as 0.

```
diff <- deviance(lmod2) - deviance(lmod)
diff
```

```
## [1] 328.8398
```

```
p_val <- 1 - pchisq(diff, 2)
sprintf("%.10f", p_val)
```

```
## [1] "0.0000000000"
```

2.e

Now let's compare based on AIC, using the *step* command, starting from the largest possible model *lmod*

```
# required to get step to
# work without error
pima_omit_na <- na.omit(pima)
lmod <- glm(test ~ pregnant + glucose + diastolic + triceps
          + insulin + bmi + diabetes + age, family=binomial, pima_omit_na)
lmodr <- step(lmod, trace=0)
sumary(lmodr)
```

```
##               Estimate Std. Error z value  Pr(>|z|)
## (Intercept) -9.9920797  1.0868663 -9.1935 < 2.2e-16
## pregnant     0.0839530  0.0550306  1.5256 0.1271173
```

```
## glucose      0.0364578  0.0049779  7.3240 2.407e-13
## bmi          0.0781387  0.0206053  3.7922 0.0001493
## diabetes     1.1509128  0.4242424  2.7129 0.0066704
## age          0.0343604  0.0178096  1.9293 0.0536918
##
## n = 392 p = 6
## Deviance = 344.88513 Null Deviance = 498.09781 (Difference = 153.21267)
```

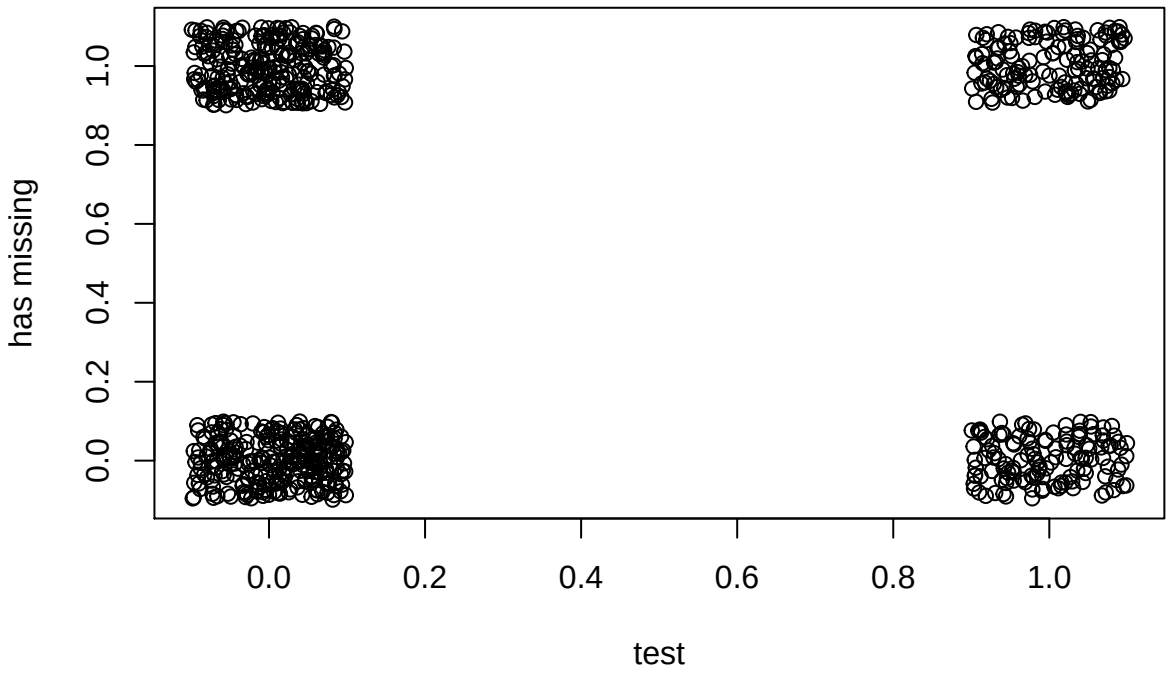
As we can see, *pregnant*, *bmi*, *diabetes*, and *age* are selected as predictors (although it's noteworthy that the p-value on *pregnant* is doesn't pass significance). 392 data points are used in the final model.

2.f

```
# let's add a column
# to the dataframe representing
# if the row has any missing values at all
pima$has_missing <- !complete.cases(pima)

# now let's regress test
# on has_missing
lmod_missing_values <- glm(
  test ~ as.numeric(has_missing),
  family=binomial, pima)

sumary(lmod_missing_values)
```

```
##                        Estimate Std. Error z value  Pr(>|z|)
## (Intercept)             -0.70081    0.10728 -6.5325 6.468e-11
## as.numeric(has_missing)  0.15579    0.15152  1.0282    0.3038
##
## n = 768 p = 2
## Deviance = 992.42598 Null Deviance = 993.48391 (Difference = 1.05793)
```

```
# plotting this relationship,
# we can see visually there isn't much of a
# correlation between the two variables
plot(jitter(as.numeric(has_missing), .5) ~ jitter(test, .5), xlab="test", ylab="has missing", pima)
```



As we can see above, the coefficient on *has_missing* in the regression is slight and has a p-value of .3, indicating we should not reject the null that *test* and *has_missing* are uncorrelated.

See above for a jittered plot of *has_missing* against *test*. Because *has_missing* is uncorrelated with the response, it's fine to drop missing data when we run the following regression.

```
# now let's rerun the previous model on the whole dataset
lmodr2 <- glm(test ~ pregnant + glucose + bmi + diabetes + age, family = binomial, pima)
sumary(lmodr)
```

```
##               Estimate Std. Error z value  Pr(>|z|)
## (Intercept) -9.9920797  1.0868663 -9.1935 < 2.2e-16
## pregnant     0.0839530  0.0550306  1.5256 0.1271173
## glucose      0.0364578  0.0049779  7.3240 2.407e-13
## bmi          0.0781387  0.0206053  3.7922 0.0001493
## diabetes     1.1509128  0.4242424  2.7129 0.0066704
## age          0.0343604  0.0178096  1.9293 0.0536918
##
## n = 392 p = 6
## Deviance = 344.88513 Null Deviance = 498.09781 (Difference = 153.21267)
```

```
sumary(lmodr2)
```

```
##               Estimate Std. Error  z value  Pr(>|z|)
## (Intercept) -9.3227891  0.7372791 -12.6449 < 2.2e-16
## pregnant     0.1150579  0.0323411   3.5576 0.0003742
## glucose      0.0359411  0.0035551  10.1097 < 2.2e-16
## bmi          0.0875291  0.0147224   5.9453  2.76e-09
## diabetes     0.9205827  0.3008320   3.0601 0.0022125
## age          0.0113658  0.0093151   1.2202 0.2224047
##
## n = 752 p = 6
## Deviance = 703.24121 Null Deviance = 974.74647 (Difference = 271.50526)
```

2.g

Let's compute the *odds_ratio*, which captures the "difference" (as a ratio) between the odds of testing positive over two hypotheses. As well, let's use a confidence interval on β_{BMI} to construct a confidence interval over the odds ratio:

```
bmi_coeff <- coef(lmodr2)[[4]]
bmi_quartiles <- quantile(pima$bmi, na.rm=TRUE)
bmi_quartiles
```

```
##   0%  25%  50%  75% 100%
## 18.2 27.5 32.3 36.6 67.1
```

```
third_quartile <- bmi_quartiles[[3]]
first_quartile <- bmi_quartiles[[1]]
odds_ratio <- exp(third_quartile * bmi_coeff) / exp(first_quartile * bmi_coeff)

# here's the ratio of the odds of testing positive
# with BMI at the third quartile vs the first quartile
odds_ratio
```

```
## [1] 3.435494
```

```
bmi_confint <- confint(lmodr2, "bmi")
```

```
## Waiting for profiling to be done...
```

```
lowerbound <- bmi_confint[[1]]
upperbound <- bmi_confint[[2]]

upperbound_odds_ratio <- exp(third_quartile * upperbound) / exp(first_quartile * upperbound)

lowerbound_odds_ratio <- exp(third_quartile * lowerbound) / exp(first_quartile * lowerbound)

# lowerbound of confidence interval
lowerbound_odds_ratio
```

```
## [1] 2.304449
```

```
# upperbound of confidence interval
upperbound_odds_ratio
```

```
## [1] 5.205946
```

2.h

Let's compute the means of diastolic conditional on *test*, and at the population level:

```
# population level mean
m <- mean(pima$diastolic, na.rm = TRUE)
m
```

```
## [1] 72.40518
```

```
# mean conditional on test == 0
pima_test_negative <- pima[pima$test == 0,]
m_test_neg <- mean(pima_test_negative$diastolic, na.rm = TRUE)
m_test_neg
```

```
## [1] 70.87734
```

```r
# mean conditional on test == 1
pima_test_positive <- pima[pima$test == 1,]
m_test_pos <- mean(pima_test_positive$diastolic, na.rm = TRUE)
m_test_pos
```

```
## [1] 75.32143
```

Now let's summarize the model:

```r
sumary(lmod2)
```

```
##               Estimate Std. Error  z value  Pr(>|z|)
## (Intercept) -8.9621457  0.8208917 -10.9176 < 2.2e-16
## pregnant     0.1178628  0.0334181   3.5269 0.0004204
## glucose      0.0351944  0.0036049   9.7630 < 2.2e-16
## diastolic   -0.0089162  0.0086176  -1.0346 0.3008356
## bmi          0.0909264  0.0157398   5.7768 7.612e-09
## diabetes     0.9605147  0.3064152   3.1347 0.0017204
## age          0.0169443  0.0098344   1.7230 0.0848936
##
## n = 724 p = 7
## Deviance = 672.86106 Null Deviance = 931.93733 (Difference = 259.07627)
```

From comparing means in the dataset, we see that women who test positive do have higher diastolic blood pressure than those who test negative (and in fact, the population at large). However, the coefficient on *diastolic* is not significant in this model, suggesting that *diastolic* does not affect *test*.

This seems like a contradiction – from simply comparing empirical means, we see that *test* and *diastolic* covary. However, significance is distinct from the sign and magnitude of the coefficient. The estimated coefficient on *diastolic* tells us "what" is happening in the observed data, but the p-value tries to get at "why" we've observed it.

More precisely, the p-value tells us how likely the estimated coefficient would be under the null; we can detect an empirical correlation between *test* and *diastolic*, but the p-value tells us the probability that such a coefficient would be generated by pure noise.