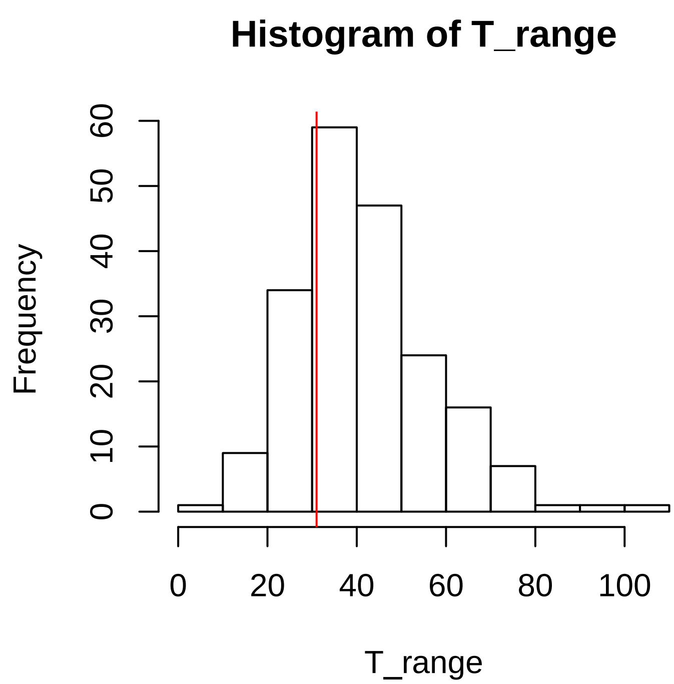
STAT 8700 Homework 5

Brian Detweiler

Friday, September 30, 2016

1. For the Schools data, check the model using the test statistic $T(y) = max_j y_j - min_j y_j$, the range. Calculate the p-value for this posterior predictive check.

```
yrep <- rnorm(8 * 200, sim.theta, schooldata.sigmaj)
yrep <- matrix(yrep, 200, 8)

diff.range <- function(row) {
  return(diff(range(row)))
}

T_range <- apply(yrep, 1, diff.range)
p.val <- 1 - sum(T_range < diff(range(schooldata.y))) / 200
hist(T_range)
abline(v = diff(range(schooldata.y)), col = "red")
```



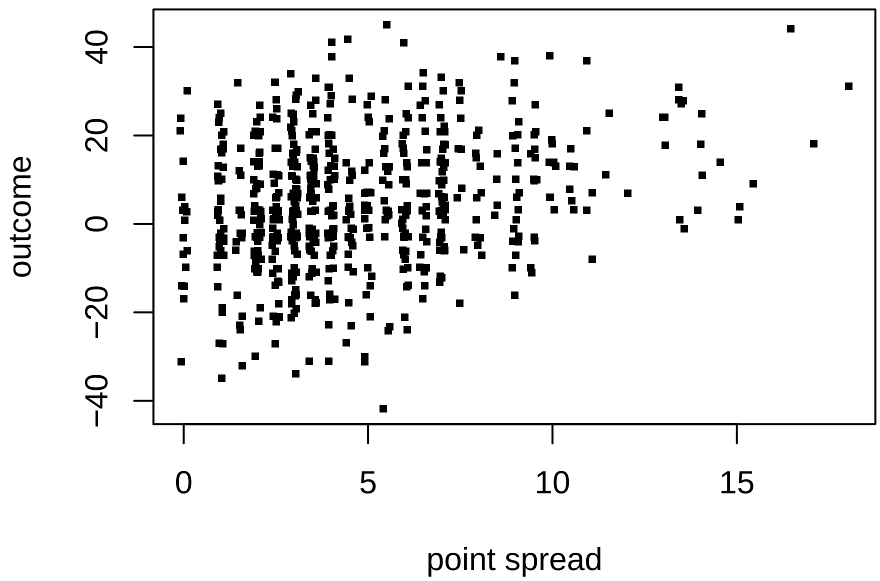
This produces a p-value of 0.755, which is well within the observed data.

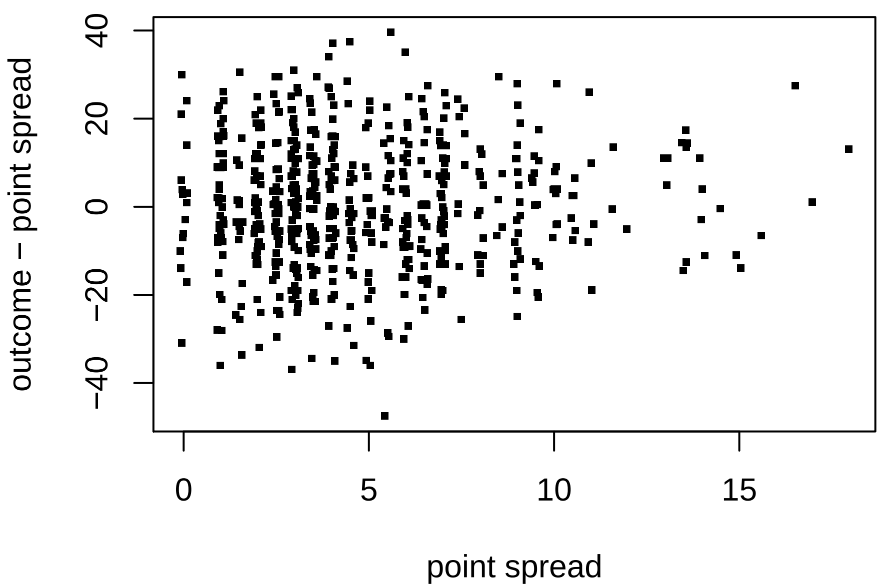
■

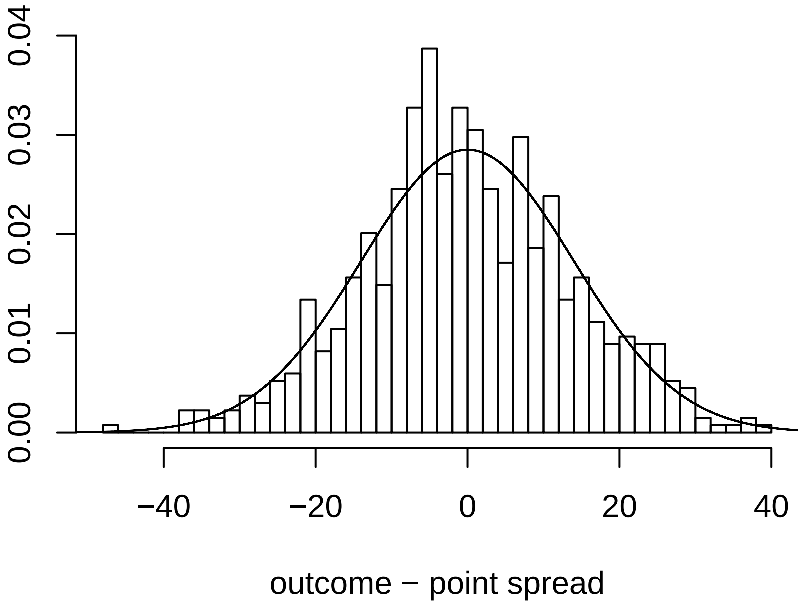
2. Read through the model for Football Points Spreads in Section 1.6. The model de-scribed in chapter 1 is of the form $y \sim Normal(x, 14^2)$ implying that $y - x \sim Normal(0, 14^2)$, however figure 1.2a seems to show a pattern of decreasing variance of $y - x$ as a function of x. The data can be found in football.txt on Blackboard, and can be read into R using `read.table("football.txt", header=T)`.

```
football.data <- read.table('football.txt', header=T)
football.data$outcome <- football.data$favorite - football.data$underdog
football.data$outcome.minus.pointspread <- football.data$outcome - football.data$spread
head(football.data)
```

```
##   home favorite underdog spread favorite.name underdog.name week outcome
## 1    1       21       13    2.0            TB           MIN    1       8
## 2    1       27        0    9.5           ATL            NO    1      27
## 3    1       31        0    4.0           BUF           NYJ    1      31
## 4    1        9       16    4.0           CHI            GB    1      -7
## 5    1       27       21    4.5           CIN           SEA    1       6
## 6    0       26       10    2.0           DAL           WAS    1      16
##   outcome.minus.pointspread
## 1                       6.0
## 2                      17.5
## 3                      27.0
## 4                     -11.0
## 5                       1.5
## 6                      14.0
```

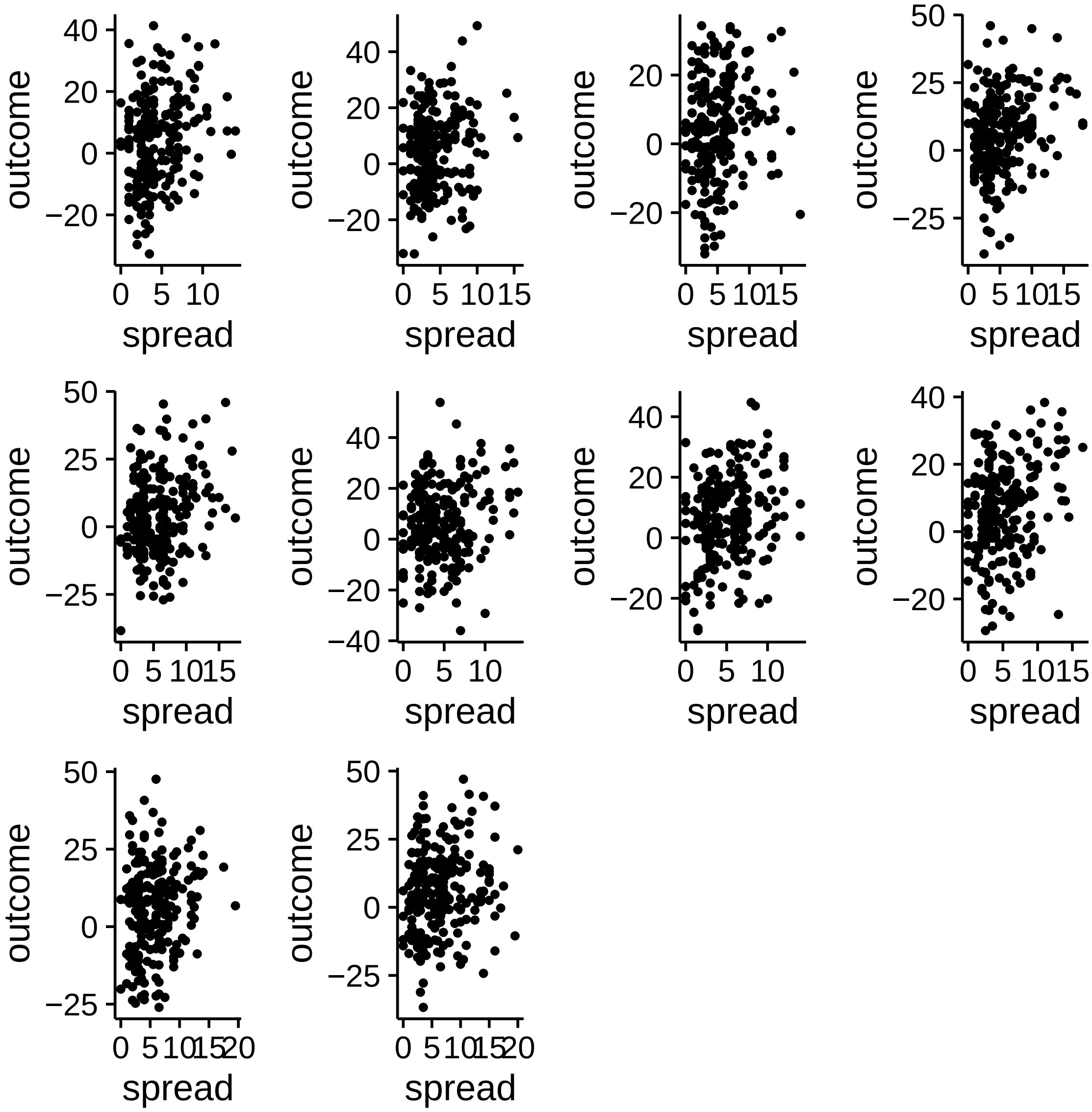

outcome
40
20
0
−20
−40
0
5
10
15
point spread


outcome − point spread
40
20
0
−20
−40
0
5
10
15
point spread

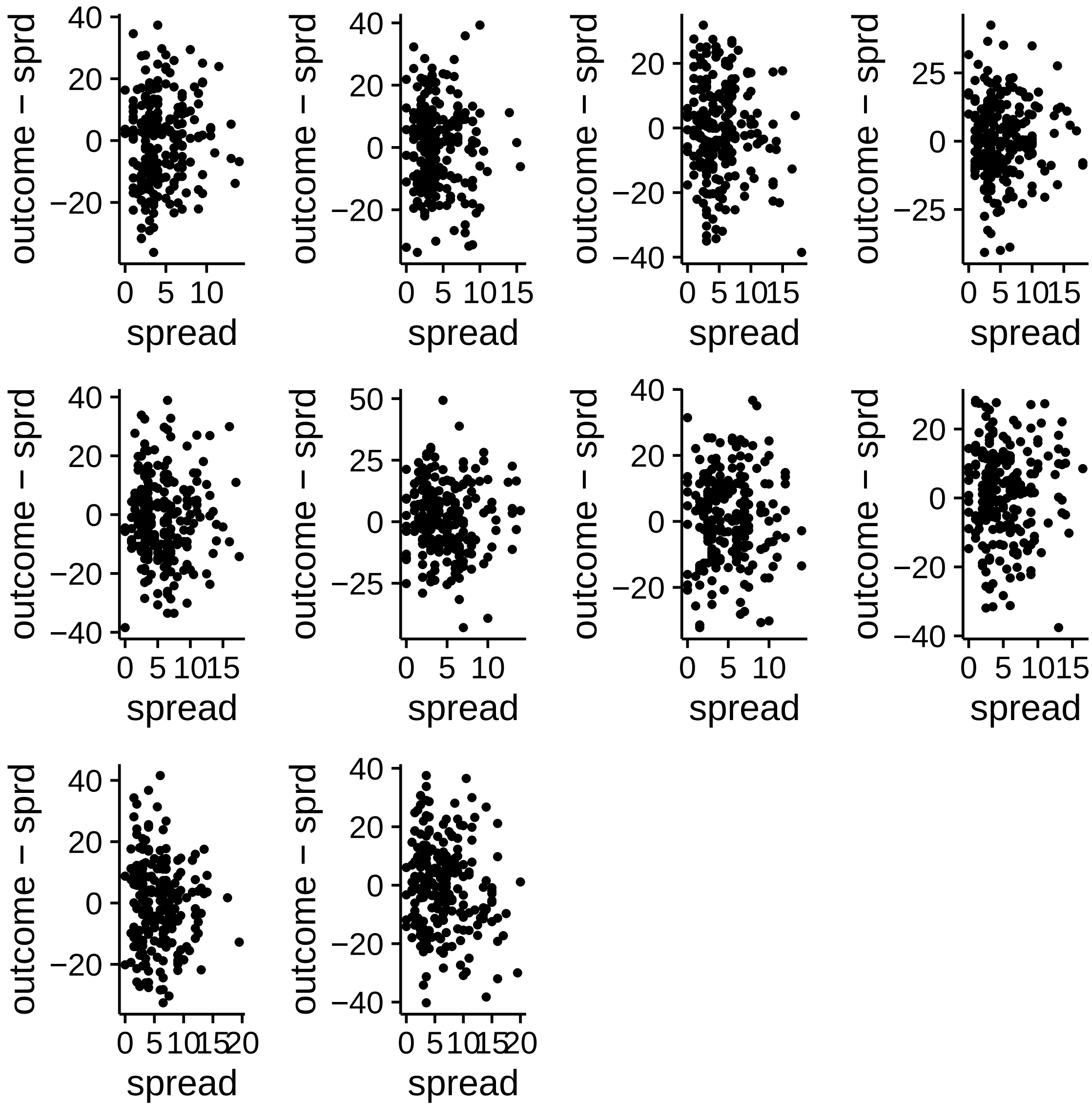

0.00
0.01
0.02
0.03
0.04
−40
−20
0
20
40
outcome − point spread

(a) Simulate several replicated data sets y^{rep} under the model and, for each, create graphs like Figurers 1.1 and 1.2. Display several graphs per page, and compare these to the corresponding graphs of the actual data. This is a graphical posterior predictive check as described in Section 6.4

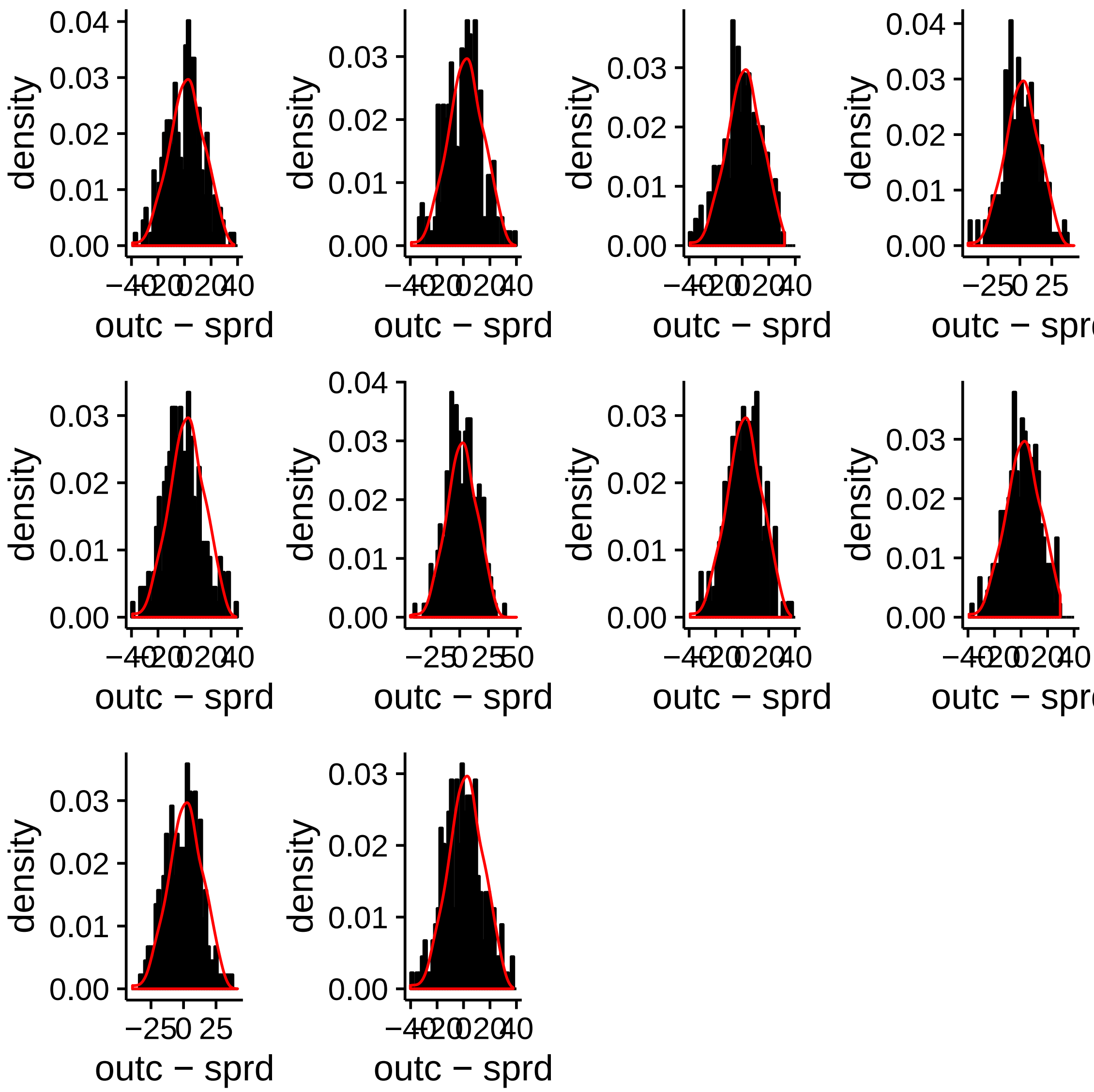
```
require(cowplot)


# Simulate y's using the data from the x's
xrep <- football.data$spread
yrep <- rnorm(length(xrep), xrep, 14)

df.list <- vector("list", 10)

for (i in 1:10) {

  from <- ((i - 1) * 224) + 1

  to <- i * 224

  dens <- rnorm(n = 224, 0, 14)
  df.list[[i]] <- as.data.frame(cbind(xrep[from:to],
                                      yrep[from:to],
                                      yrep[from:to] - xrep[from:to],
                                      dens),
                                row.names = c(1:224))
  colnames(df.list[[i]]) <- c('spread', 'outcome', 'outcome.minus.spread', 'density')
}

plot11.list <- vector("list", 10)
for (i in 1:10) {
  plot11.list[[i]] <- ggplot(df.list[[i]], aes(spread, outcome)) +
                        geom_point(size = 1) +
                        labs(x = 'spread',
                             y = 'outcome')
}

plot_grid(plotlist = plot11.list)
```




```
plot12a.list <- vector("list", 10)
for (i in 1:10) {
  plot12a.list[[i]] <- ggplot(df.list[[i]], aes(spread, outcome.minus.spread)) +
                       geom_point(size = 1) +
                       labs(x = 'spread',
                            y = 'outcome - sprd')
}

plot_grid(plotlist = plot12a.list)
```




```
plot12b.list <- vector("list", 10)
for (i in 1:10) {
  plot12b.list[[i]] <- ggplot(df.list[[i]], aes(x = outcome.minus.spread)) +
                          geom_histogram(aes(y =..density..),
                                         breaks = seq(-40, 40, by = 2),
                                         col = "black",
                                         alpha = .2) +
                          geom_density(aes(x = dens),
                                       col = "red") +
                          labs(x = 'outc - sprd')
}

plot_grid(plotlist = plot12b.list)
```

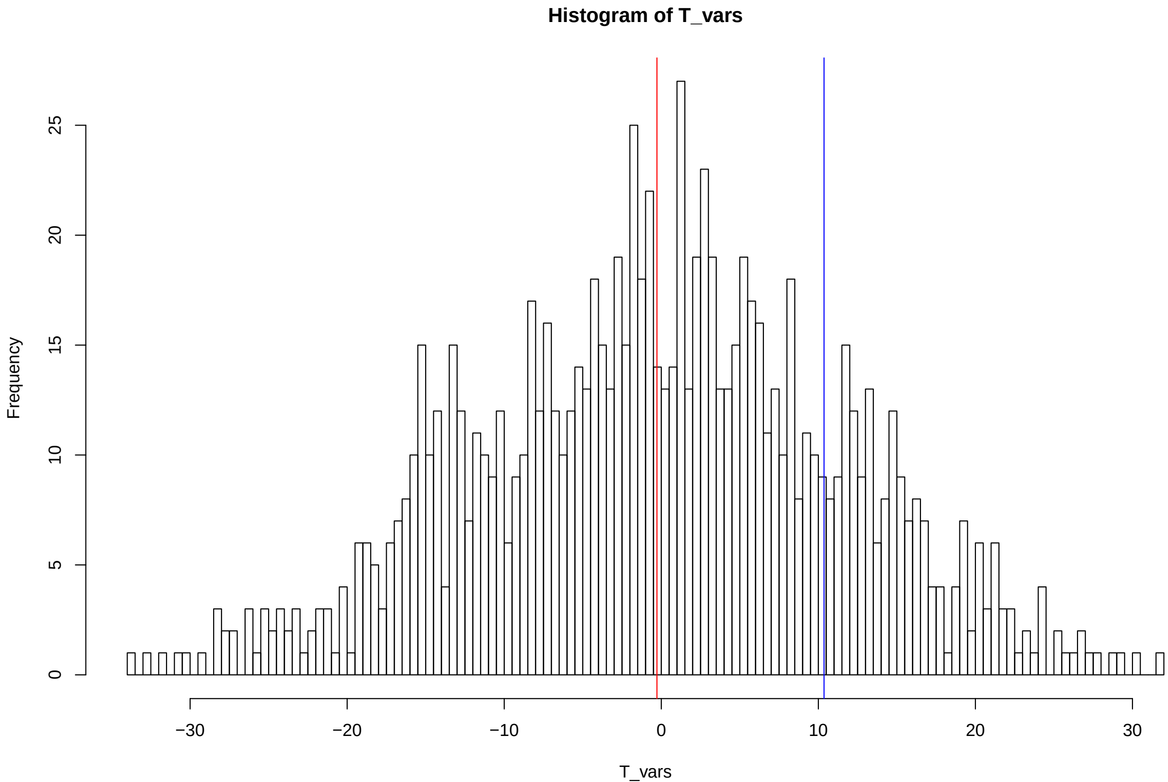

density
0.00
0.01
0.02
0.03
0.04
outc − sprd
−25
0
25
50

■

(b) Create a numerical summary $T(x, y)$ to capture the apparent decrease in variance of $y - x$ as a function of x. Compare this to the distribution of simulated test statistics, $T(x, y^{rep})$ and compute the p-value for this posterior predictive check.

The test statistic will compute the variance of two halves of the data. We will split the data along the median of the point spreads of the original data, which is 4.5. Then we subtract the lower from the upper. If the variance is generally decreasing with a larger point spread, we should see a positive mean.

```
T_var <- function(data) {
  lower <- data$outcome.minus.pointspread[data$spread <= 4.5]
  upper <- data$outcome.minus.pointspread[data$spread > 4.5]
  T_var <- var(lower) - var(upper)
}

xrep <- football.data$spread

temp <- football.data

T_vars <- c()

# 1000 simulations of outcomes given point spreads
for(i in 1:1000) {
  yrep <- rnorm(length(xrep), xrep, 14)
  temp$outcome <- yrep
  temp$outcome.minus.pointspread <- yrep - xrep
  T_vars <- c(T_vars, T_var(temp))
}

original.T_var <- T_var(football.data)

hist(T_vars, breaks=99)
abline(v = mean(T_vars), col = 'red')
abline(v = T_var(football.data), col = 'blue')
```




```
p.val <- 1 - sum(T_vars < original.T_var) / 1000
```

The mean of the simulations is -0.2755219 (plotted in red). This is closer to normal, which we might expect having used a normal with mean 0. The original T_{var} value for the data is 10.3631893. This is plotted in blue.

The p-value is 0.189, which is still within our acceptable range, but as the book says, this model is not perfect.

■