

Computer Lab 5

Andrea Bruzzone, Thomas Zhang

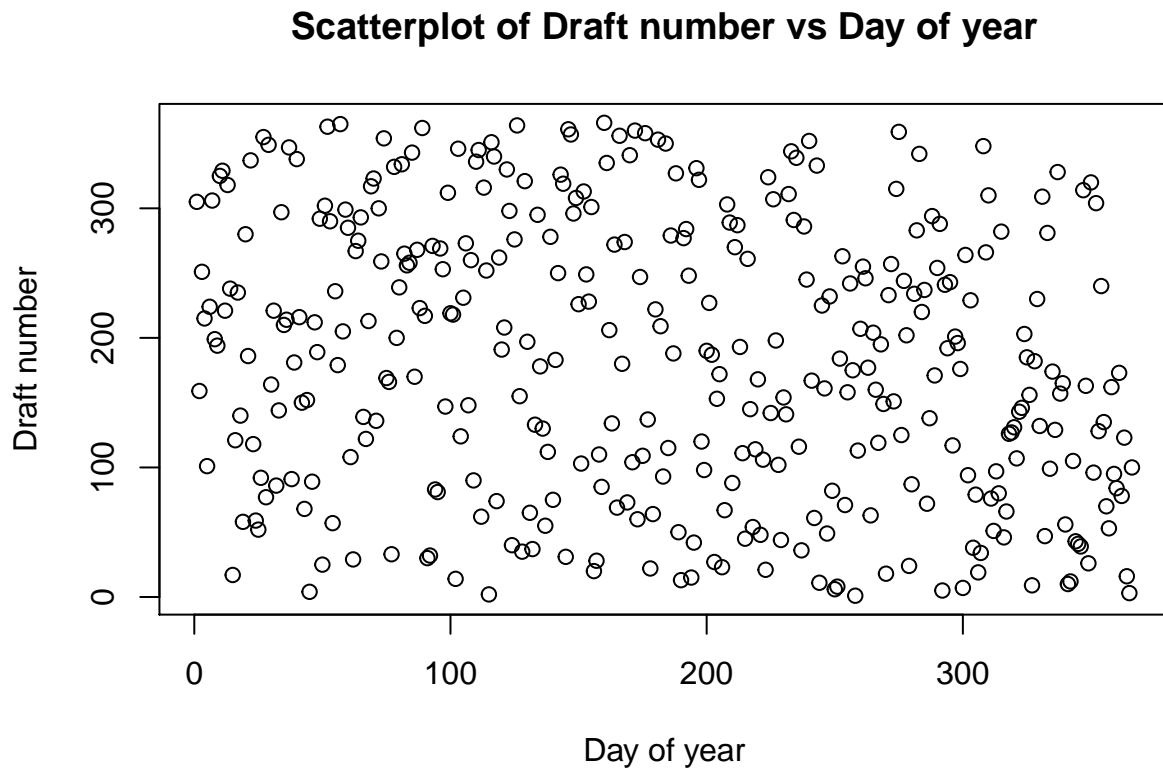
2016 M03 9

Assignment 1

In this assignment we want to investigate if the draft number given in a random selection process for the military draft is truly random. Thus, the Y =Draft Number and the X =Day of year.

- 1.1

This is the scatterplot of Y versus X :



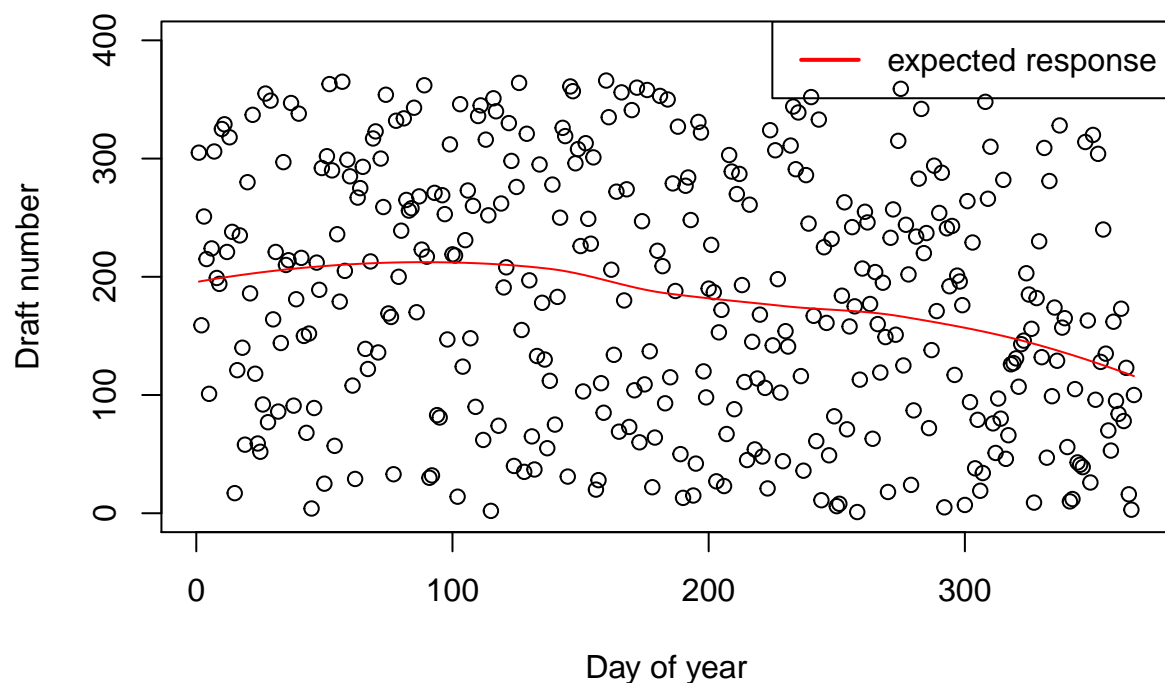
From the scatterplot it can be said that the lottery looks random.

- 1.2

An estimate of the expected response as a function of X is computed using the `loess()` function.

Then, we plot the curve obtained in the previous graph:

Scatterplot of Draft number vs Day of year



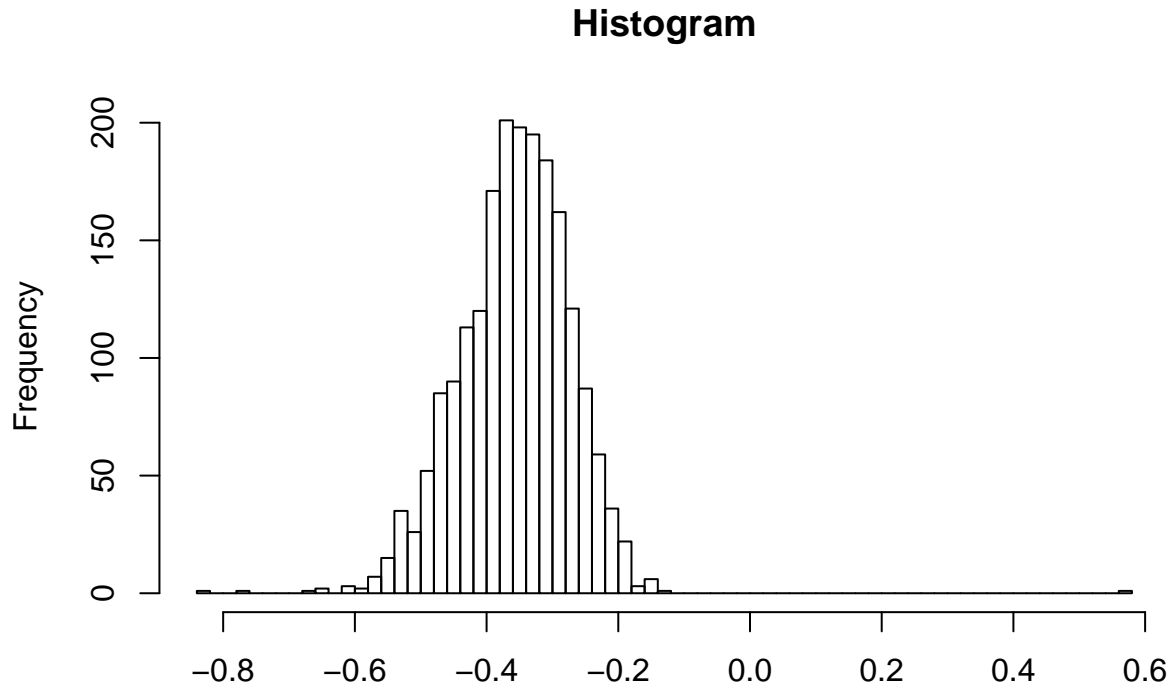
Again, it seems that the lottery still looks random.

- 1.3

To check if the lottery is random, we use this statistic:

$$T = \frac{\hat{Y}(X_b) - \hat{Y}(X_a)}{X_b - X_a} \text{ where } X_b = \operatorname{argmax}_X Y, X_a = \operatorname{argmin}_X Y$$

If this value is significantly greater than zero the lottery is not random. We estimate the distribution of T by using a non parametric bootstrap with B=2000.



From the histogram, the T statistic seems to be a t-student.

Moreover, it can be seen that all the values are smaller than 0, for this reason if we compute the p-value that should lead to reject the hypothesis that T is greater than zero.

The p-value is:

```
## [1] 5e-04
```

The p-value is very small, almost zero for this reason with a test at 5% we refuse the null hypothesis like we said before and we can say that the lottery is random.

Let us do further hypothesis testing by using a permutation test, where we permute the day of year variable, then calculate test statistic T . We replicate this procedure 2000 times and then take the p-value to be the proportion of T 's greater than the empirical value of the test statistic, in this case the empirical value is $T_0 = -0.3479$.

```
## [1] "p-value from permutation testing: 0.924"
```

We discover that the p-value is far too high to reject the null hypothesis. The fact that the p-value is higher than one half could possibly be explained by the fact that the empirical observation of T is a negative number.

Let us try to crudely estimate the power of this permutation test by creating non-random (alternative hypothesis) "draft number" data $Y(x)$ using the day of year data column x of the form

$$Y(x) = \max(0, \min(\alpha x + \beta, 366)) \quad \text{where}$$

$$\alpha = 0.1 \quad \text{and} \quad \beta \sim N(183, 100)$$

Since this data is non-random, the permutation test function as written by me should return very low p-value.

```
## [1] "p-value from permutation testing non-random Draft_No, alpha = 0.1: 0"
```

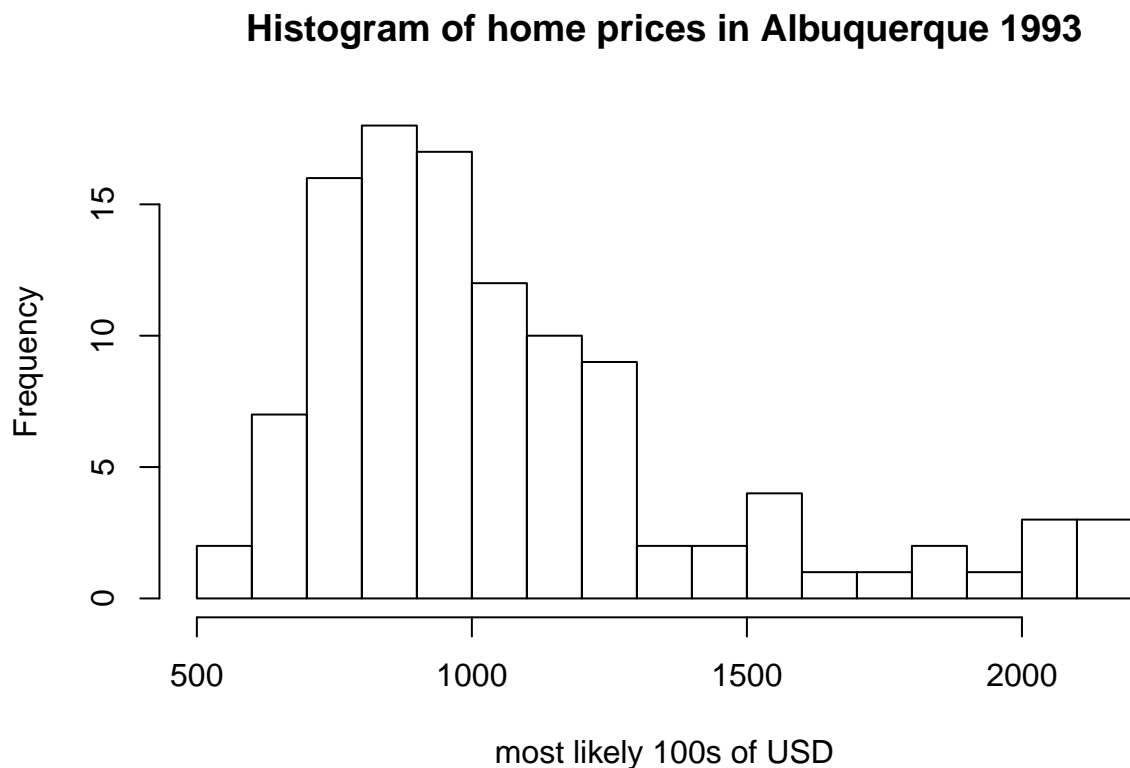
It does look as if that would be the case. We continue this line of exploration by setting α to 0.1, 0.2, ..., 10 and see when the permutation test will reject the null hypothesis (data is random) at 5% significance level. We then take the proportion of (absolutely correct) rejections to be the crude estimate of the power of the test.

```
## [1] "Crude estimate of Power of the test statistic T: 1"
```

The power estimated would indicate that this is a good test for non-randomness. However, I would like to point out that above $\alpha = 0.5$ increasingly the Draft numbers take on the maximum value of 366 and that might affect the result.

Assignment 2

We have the home prices in Albuquerque 1993 and we plot a histogram and find the mean home price.

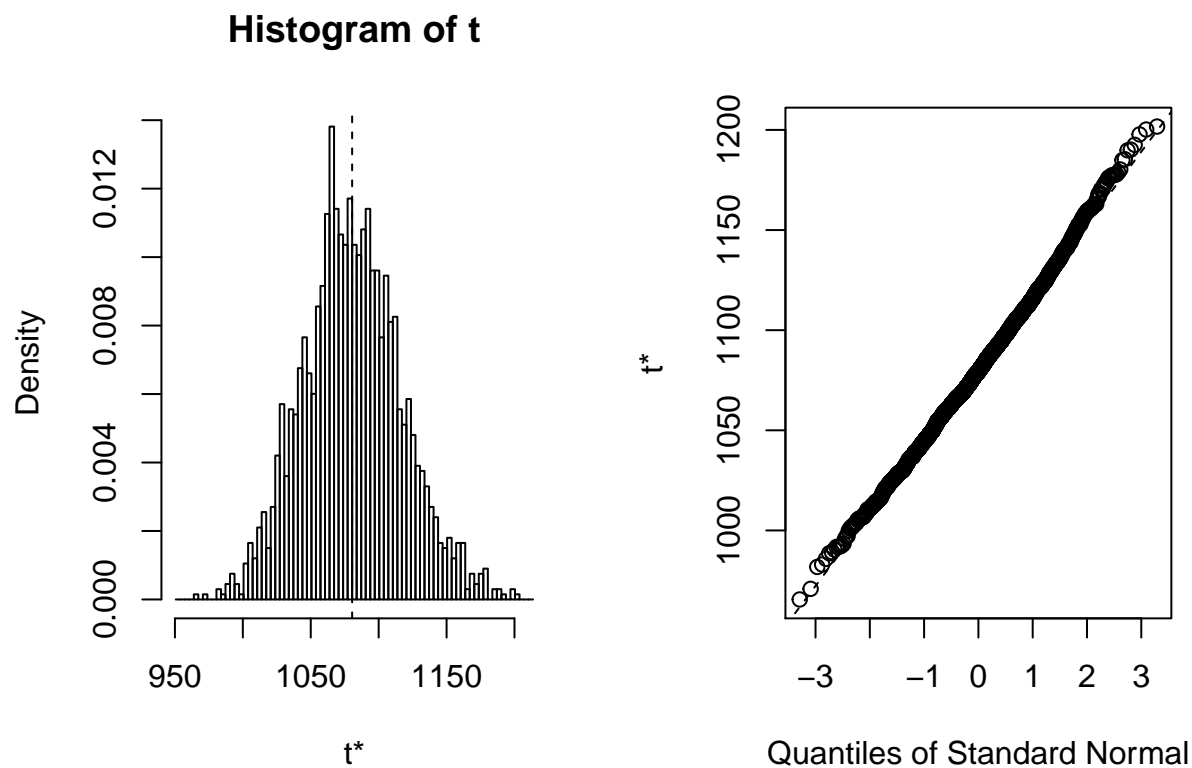


```
## [1] "mean price of home in Albuquerque 1993: 1080.473 hundred USD"
```

I can not say for sure, but I believe the price unit is hundreds of USD. The distribution of home prices look a little like a chi-squared distribution.

We are going to estimate the distribution of the mean home price using bootstrap. We run bootstrap once to generate 2000 mean home prices, we estimate the bias-correction for the bootstrap mean house price and

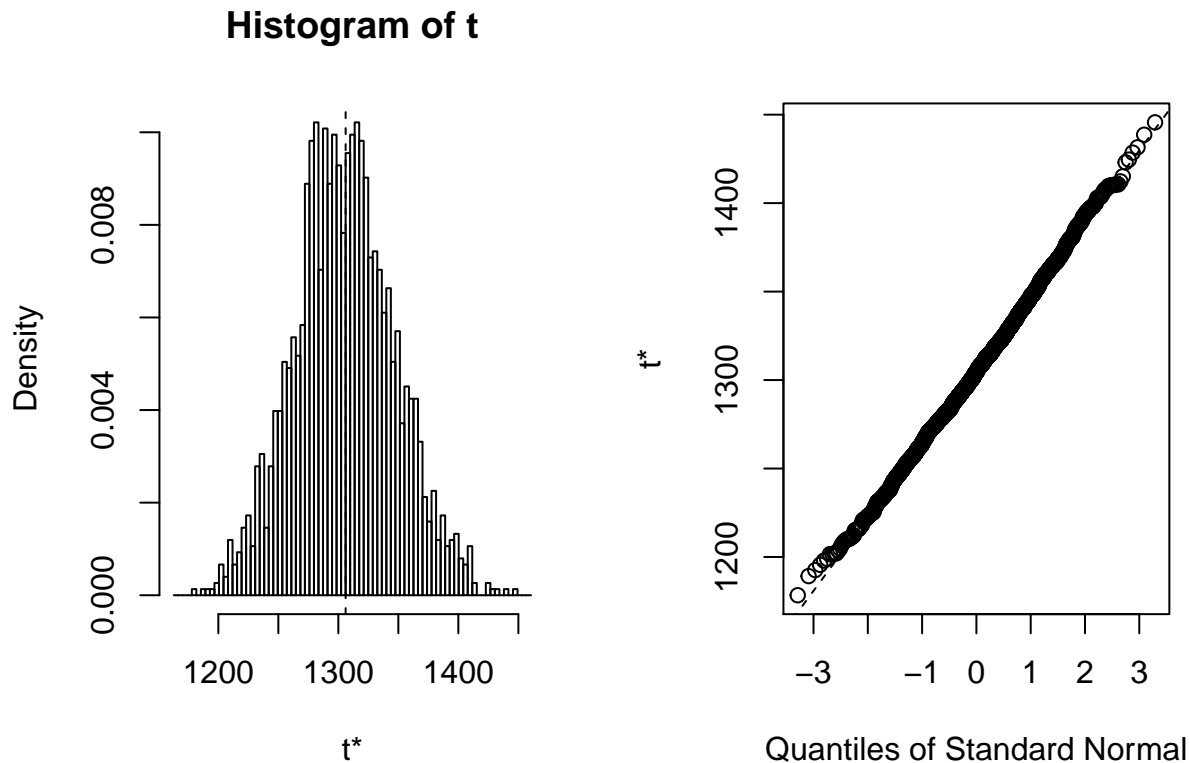
then we are going to use those bootstrap mean home prices to find the estimate of the variance of bootstrap mean house prices, using 2000 bootstrap replicates.



```
## [1] "Bootstrap mean house price bias-correction estimate -0.21 hundred USD"
```

```
## [1] "Bias-corrected bootstrap estimate of mean house price: 1080.262 hundred USD"
```

We can see that the bootstrap mean house prices are almost perfectly normally distributed and the bias in mean house price is very small.



```
## [1] "Bootstrap estimate of variance of mean house price: 1305.305 hundred USD squared"
```

We can compare the estimate of the variance of mean house price to an estimate of the variance of mean house price obtained using the jackknife method.

```
## [1] "Estimate of variance of mean home price by jackknife: 1320.911 hundred USD squared"
```

We find that the variance estimate generated by the jackknife method is slightly higher than the bootstrap variance estimate.

Now we create 95% bootstrap confidence intervals for the mean house price using the normal approximation method, the percentile method and the BCa method and compile the data in a table.

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot2, type = c("norm", "perc", "bca"))
##
## Intervals :
## Level      Normal          Percentile      BCa
## 95% (1009, 1151 ) (1012, 1158 ) (1016, 1162 )
## Calculations and Intervals on Original Scale
```

		method	lower	upper	length	midpoint
## 1		Normal Approximation	1009.431	1151.094	141.6636	1080.262
## 2	bootstrap percentile method		1011.916	1157.789	145.8723	1084.852
## 3		BCa method	1015.600	1161.730	146.1294	1088.665

We find that as one goes down the table the confidence interval shifts slightly higher. Interestingly, the normal approximation midpoint is the same value as the bias-corrected bootstrap estimate of the mean house price. (Possibly this is because the bootstrap home mean prices are almost normally distributed)

Group contribution

Andrea provided the first three tasks in code and text and Thomas provided the rest. Of course, we helped each other with the knowledge parts where we respectively felt at a disadvantage. All aspects of hypothesis testing and bootstrap have been thoroughly discussed.

Appendix

R code

```
library(bootstrap)
library(boot)
library(XLConnect)
wb = loadWorkbook("lottery.xls")
wb2 = loadWorkbook("prices1.xls")
data = readWorksheet(wb, sheet = "Sheet1", header = TRUE)
#data is changed around alot in data$Draft_No. reset often.
data2 = readWorksheet(wb2, sheet = "Sheet1", header = TRUE)

t <- loadWorkbook("lottery.xls")
lottery <- readWorksheet(t, sheet = "Sheet1", header = TRUE)

#1.1
plot(lottery$Day_of_year, lottery$Draft_No, ylab="Draft number", xlab = "Day of year",
     main = "Scatterplot of Draft number vs Day of year")

lo <- loess(Draft_No ~ Day_of_year, lottery)
pred <- predict(lo)

plot(lottery$Day_of_year, lottery$Draft_No, ylab="Draft number", xlab = "Day of year",
     main = "Scatterplot of Draft number vs Day of year", ylim = c(0, 400))
lines(lottery$Day_of_year, pred, col = "red")
legend("topright", c("expected response"),
      lty=c(1,1), lwd = c(2, 2), col = "red")
D <- 2000
stat <- numeric(D)
n <- dim(lottery)[1]
set.seed(-3447)
for(d in 1:D){
  data1 <- lottery[sample(nrow(lottery), nrow(lottery), replace = TRUE), ]
  los <- loess(Draft_No~Day_of_year, data1)
  new_data1 <- data.frame(X = data1$Day_of_year, Y = los$fitted)
```

```

mx <- which.max(new_data1$Y)
mi <- which.min(new_data1$Y)
stat[d] <- (new_data1$Y[mx] - new_data1$Y[mi]) / (new_data1$X[mx] - new_data1$X[mi])
}

hist(stat, 50, main = "Histogram", xlab = "", breaks=100)
pval <- mean(stat>0)
pval
permfun <- function(data,B){
  n <- dim(data)[1]
  permvec <- rep(0,B)
  for(i in 1:B){
    rando <- sample(1:n,n)
    dat2 <- data.frame(day = rando, draft = data$Draft_No)
    poly <- loess(draft ~ day, data = dat2)
    smallest <- which.min(poly$fitted)
    largest <- which.max(poly$fitted)
    TEE <- (poly$fitted[largest] - poly$fitted[smallest]) /
      (dat2$day[largest] - dat2$day[smallest])
    permvec[i] <- TEE
  }
  poly <- loess(Draft_No ~ Day_of_year, data = data)
  smallest <- which.min(poly$fitted)
  largest <- which.max(poly$fitted)
  originalstatistic <- (poly$fitted[largest] - poly$fitted[smallest]) /
    (data$Day_of_year[largest] - data$Day_of_year[smallest])
  prop <- mean(permvec > originalstatistic)
  return(prop)
}
set.seed(-3447)
pval2 <- permfun(data,B= 2000)
paste("p-value from permutation testing: ",pval2)
alpha <- 0.1

nonrandgen <- function(data,alpha){
  nonrandY <- numeric(366)
  for(i in 1:366){
    beta <- rnorm(1,183,10)
    nonrandY[i] <- max(0,min(alpha * data$Day_of_year[i] + beta, 366))
  }
  return(nonrandY)
}
#look, above alpha = 0.5 we get the ceiling alot.
# what is the point?
#num, forgot about the permutation of days

nonrandY <- nonrandgen(data,alpha)
data$Draft_No <- nonrandY
pval3 <- permfun(data,B = 200)
paste("p-value from permutation testing non-random Draft_No, alpha = 0.1: ",pval3)
alpha <- seq(from = 0.1, to = 10 , by = 0.1)

pvals <- rep(0,length(alpha))

```



```

for(j in 1:length(alpha)){
  nonrandY <- nonrandgen(data,alpha[j])
  data$Draft_No <- nonrandY
  pvalcurr <- permfun(data, B = 200)
  pvals[j] <- pvalcurr
}
paste("Crude estimate of Power of the test statistic T: ", round(mean(pvals < 0.05),2))
hist(data2$Price,breaks = 20,main="Histogram of home prices in Albuquerque 1993",
      xlab="most likely 100s of USD")
paste("mean price of home in Albuquerque 1993: ",round(mean(data2$Price),3),"hundred USD")
m <- length(data2$Price)

meanstat <- function(data,ind){
  datar <- data[ind,]
  res <- mean(datar$Price)
  return(res)
}
set.seed(-3447)
boot2 <- boot(data2,meanstat, R = 2000)
#hist(boot2$t,main="Histogram of bootstrap mean house prices",xlab = "mean house price")
plot(boot2)

#bias correction

biascorrectedest <- 2 * mean(data2$Price) - mean(boot2$t)
paste("Bootstrap mean house price bias-correction estimate",
      round(mean(data2$Price) - mean(boot2$t),3)," hundred USD")
paste("Bias-corrected bootstrap estimate of mean house price: ",
      round(biascorrectedest,3)," hundred USD")
varstat <- function(data,ind){
  datar <- data[ind,]
  res <- var(datar)
  return(res)
}
set.seed(-3447)
boot3 <- boot(boot2$t,varstat, R = 2000)
plot(boot3)
paste("Bootstrap estimate of variance of mean house price: ",
      round(mean(boot3$t),3)," hundred USD squared")
varbyjack <- jackknife(data2$Price,mean)
Tstarjs <- m * rep(mean(data2$Price),m) - (m-1) * varbyjack$jack.values
jackknifedT <- mean(Tstarjs) # this is the same as mean of house price...
jackvarofmeans <- sum((Tstarjs-jackknifedT)^2) / (m*(m-1)) # is the same as varbyjack$jack.se^2
paste("Estimate of variance of mean home price by jackknife: ",
      round((varbyjack$jack.se)^2,3)," hundred USD squared")
set.seed(-3447)
confintsboot <- boot.ci(boot2, type=c("norm","perc", "bca"))
print(confintsboot)
normci <- confintsboot$normal
percci <- confintsboot$percent
bcaci <- confintsboot$bca
confintinfos <- data.frame(method = c("Normal Approximation",
                                     "bootstrap percentile method","BCa method"),

```

```

lower = c(normci[2],percci[4],bcaci[4]),upper = c(normci[3],percci[5],bcaci[5]),
length = c(normci[3] - normci[2],percci[5] - percci[4],bcaci[5]-bcaci[4]),
midpoint = 1/2 * c(normci[3] + normci[2],percci[5] + percci[4],bcaci[5] + bcaci[4]))
confintinfos #as you go down the list the interval creeps higher,
#also normal approx midpoint is same as bias corrected bootstrap est.
## NA

```