STA 104 Final Project

Miguel Morales

TOPIC I
Subtopic 1: Permutation test for variance

In this analysis, we have a dataset that contains two samples. In order to compare two samples, we take a look at their measures of centrality. In this case we want to compare the samples using their variances. We want to know if the two variances from the two samples are statistically different

First we take a partial look at our data set.

numbers <dbl>	groups <fctr>
1	A
4	A
6	A
6	A
7	A
25	A
13	B
13	B
12	B
14	B

Our dataset contains two samples. One from group A and one from group B. Both samples have 6 observations.

Our assumptions for a permutation test to find the difference in variances are:

1. Random sample in groups
2. Groups are independent

Our Null hypotheses for this permutation test is that our groups have the same variance. While our alternative hypothesis is that the groups do not have the same variances. This is a two sided test.

Now that we have our two samples, we need to find the observed difference, D^obs. In this case our observed difference is Var(X1) - Var(X2).

Variance by group and D^obs:

```
                           Group 1  Group 2
Variance of both groups 72.56667 1.866667
                         [,1]
Difference in Variance 70.7
```

The first part of a permutation test is to create all of the possible permutations of the given data. For our two groups, we create all possible two group samples. It should be noted that every permutation of the two groups is equally likely to occur. In this case we will be shuffling the group names around. For every permuted group, we then find Di, the difference in variances for every ith permutation. Now we have created a list of "i" differences in variances.

Example of group shuffling:

```
 [1] A A B A B B A A B A B B
Levels: A B
```

As you can see, only the group names are being shuffled around. The numbers are in the exact same order, but now some pertain to the different groups. Each shuffling of the groups is equally likely to occur.

These are the first 6 Di differences in variance of the two shuffled groups:

```
 -24.90000  65.50000  25.96667 -14.03333  47.43333  41.76667
```

To calculate the p-value for a permutation test, we count the number of Di's that are the same or more extreme than our D^obs, and divide that number by the total number of permutations that we ran.

Calculated p-value:

```
p-value = 0.008666667
```

Based on our p-value, we can reject the null hypothesis as there is not enough evidence to suggest that the variances are the same. Alternatively, the evidence suggests that the variances of group A and group B are not equal to one another.

Topic II

Part 1.

I. Introduction

For this analysis, we are using a dataset called “crowdedness.csv”. This dataset contains 4 columns and covers different aspects of living conditions from several countries. Column 1 in this dataset is the list of countries that have been reported on by name. Column 2 is titled “Crowdedness” and is the average number of persons per room living in a home. Column 3 is labeled “fertility” and is the number of live births born per woman of a certain age per year. Lastly, column 4 is the GDP, a measure of national wealth, of the country.

Question #1: What direction of nonparametric tests will we use to analyze this data?
Question #2: Which estimate model is more accurate?
Question #3: What are the estimates of the missing values?

To answer question one, we should simply take a look at our data set and what data we have to our disposal. Are there any missing data points?

To answer question two, we will use multiple smoothing techniques that will create estimates, using our predictor. Use cross validation to select the best model.

To answer question three, we will be using the calculated best model and providing estimates for our missing values.

No knowledge about the distribution of the data was given and assumptions needed for parametric tests were not given, therefore we will be conducting these tests using nonparametric techniques.

The goal of this analysis is to find estimates for the missing values in the column “Crowdedness.”

II. Summary of Data

First, we will be taking a look at part of our dataset. This will provide us with information to answer question #1, showing the types of variables that we will be analyzing.

Partial look of “crowdedness.csv”:

Country	Crowdedness	fertility	GDP
<fctr>	<dbl>	<dbl>	<int>
ARUBA	0.7	NA	20100
AUSTRIA	0.7	1.28	31187
AZERBAIJAN	2.1	2.10	853
BAHAMAS	1.3	2.29	14462
BELGIUM	0.6	1.66	29257
BERMUDA	0.6	1.67	51991
BOLIVIA	NA	NA	878
BRAZIL	0.7	2.21	2700
BULGARIA	1.0	1.10	2533
CAMEROON	1.2	4.61	803

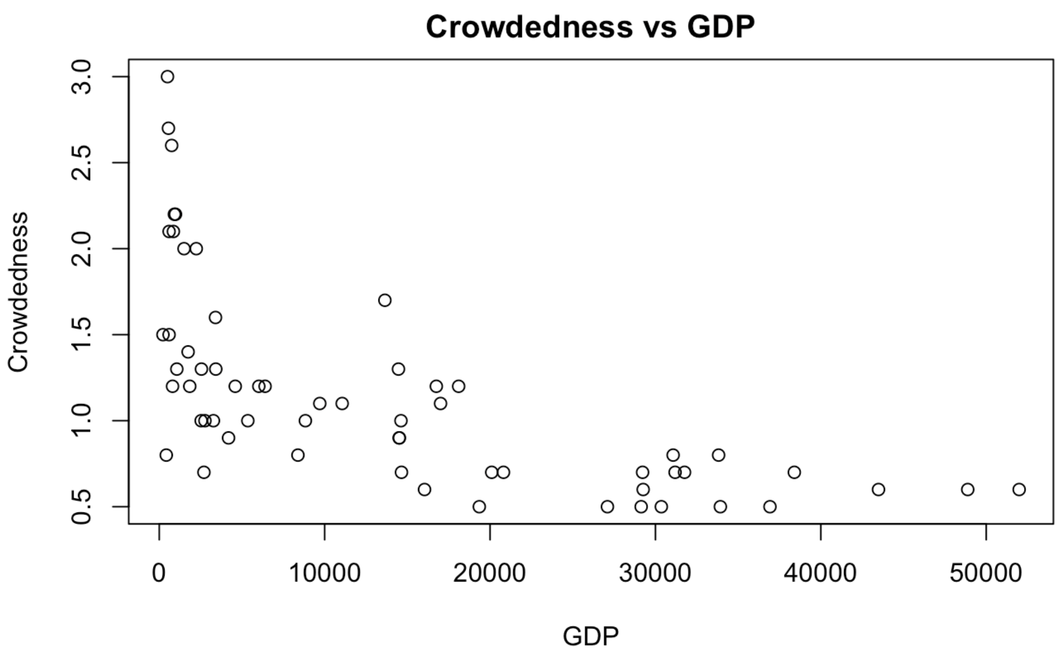
```
Country Crowdedness   fertility         GDP 
      0           2           5           0 
```

Here, we can take a look at part of our dataset to see the columns of interest. For this analysis, our goal is to estimate the missing value in the “Crowdedness" column. These missing values are labeled as “NA”. We will be estimating these missing values using GDP as our predictor variable. GDP can be used as our predictor variable because it contains no missing values. Both of these columns in the data set are numerical.

We can see that the column “Crowdedness” has a total of two missing values that we will be estimating. They are at GDP = 878 and GDP= 5945.

Since, we have no knowledge of the functional form of this data, we are approaching this analysis without knowing the distribution of our data. This entails us to use a smoothing method to estimate these missing data points. In the smoothing method, we will be using the other points in the data that are close and around the points of interest to estimate the missing observations.

Our next step is to create a plot of the variables of interest to foresee any issues with using a smoother estimate.

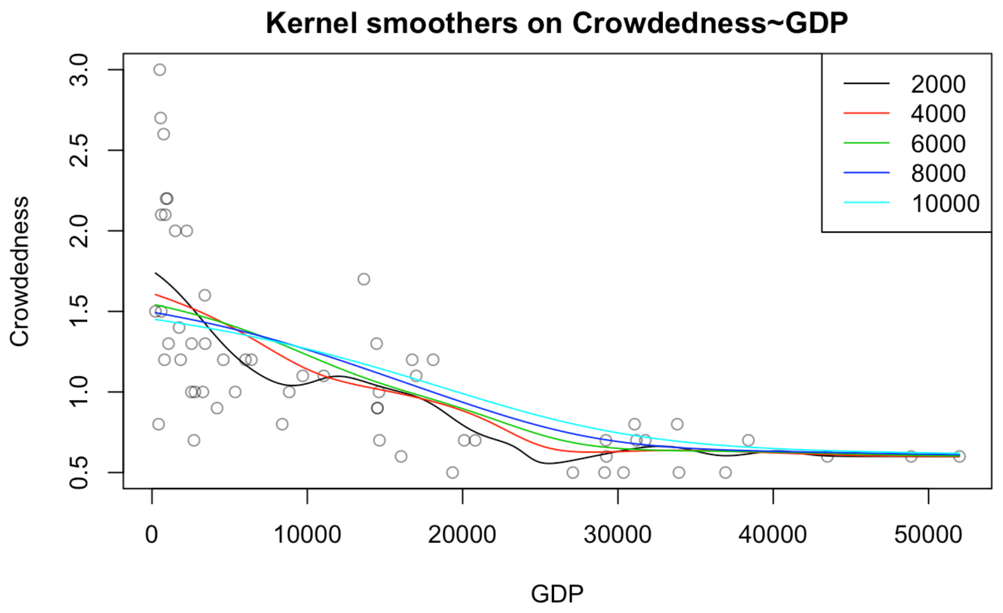


From this plot, we can see that using a smoothing estimate will come with a few issues. Since we have an estimate at 878, this is near the min of our data. Using smoothing near the min is not the most accurate way to predict Crowdedness. We also see a large variation in Crowdedness near the min of our data.

III. Analysis

The smoothing techniques that we will use to estimate the missing values are Kernel and Loess smoothing.

Kernel smoothing graph with 5 different bandwidths:



We chose bandwidths of 2000, 4000, 6000, 8000, and 10,000. The smoothing lines show that as the bandwidths increase, the lines become smoother. Our smallest bandwidth of 2000, is under smoothed, therefore will not provide accurate estimates. In order to find out which model is the best, we will be using 10-fold cross validation to find the MSPEs.

To pick the best model, we calculate the MSPEs of the models with the different bandwidths.

10-Fold Cross Validation Results:

```
             10-fold mspe
bw 2000:        391289457
bw 4000:        391289096
bw 6000:        391288648
bw 8000:        391287886
bw 10000:       391287024
```

Based on our results, we will choose the model with the lowest MSPE. We will estimate our missing values using the model with 10,000 as the bandwidth.

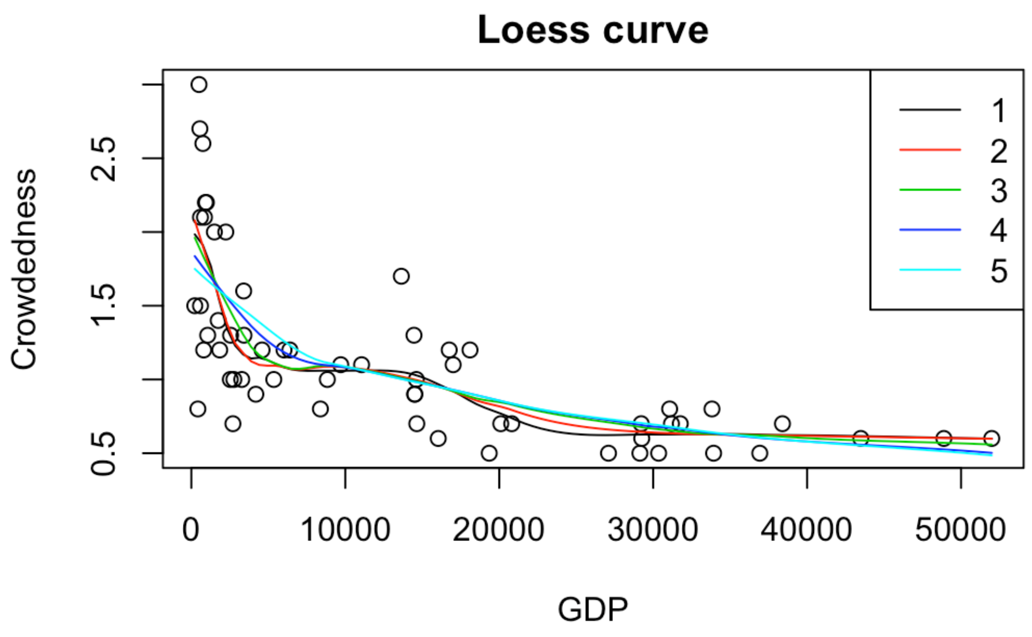
Estimation results using bw = 10,000

```
              Crowdedness Estimation
at GDP = 878                1.440526
at GDP = 5945               1.354451
```

Our estimate of crowdedness is 1.440526 at GDP = 878 and 1.354451 at GDP = 5945.

Now we do Loess smoothing and fit five different spans, .3, .4, .5 , .6, and .7.

Loess curve plot with multiple spans:



Again we choose MSPE to compare the different models.

Calculates MSPEs:

```
             MSPE
s = .3 0.1459988
s = .4 0.1516524
s = .5 0.7780417
s = .6 0.1453415
s = .7 0.1522647
```

From our results, we can see that span = 0.6 has the lowest test error. So we use this to find our estimates.

Estimates of missing values using Loess:

```
loess878  1.744550
loess5945 1.188898
```

Our Loess estimate of crowdedness is 1.744550 at GDP = 878 and 1.188898 at GDP = 5945.

IV. Interpretation

Based on our data and using GDP as our predictor variable, our kernel estimates show that the average number of persons per room living in homes in Bolivia is 1.440526. We also can say that the average number of persons per room living in homes in Mexico is 1.354451. Our Loess estimates show that the average number of persons per room living in homes in Bolivia is 1.744550. Again we also estimated that the average number of persons per room living in homes in Mexico is 1.188898.

V. Conclusion

Generally, the predictions of the smoothing types are a bit different. This is mostly due to how they weigh the different points. We can see that for Loess smoothing, towards the min of the data, the estimates are significantly higher than Kernels as it deals with the spread of the data at the min differently. However, since smoothing methods are not built using meaningful parameters, it is difficult to conclude that the estimates are entirely accurate. Especially for our estimate at 878, since it is near the min of our data where the observations are very spread out, rendering the estimate questionable using the smoothing method.

Part 2

I. Introduction

In this analysis we will again be looking at "crowdedness.csv". However we will be analysing the relationship between two of the columns. We want to know if and what is the relationship between GDP and fertility. GDP is the measure in wealth of the nations. The fertility column is the number of live births per woman of a certain age per year.

The questions that we will be answering in this analysis are as follows:
Question #1: What direction of nonparametric tests will we use to analyze the relationship between GDP and fertility?
Question #2: What is the relationship between the two models?

To answer question #1, we should simply take a look at our data set and see what type of data we have and understand any information that was given to us about this dataset.

To answer question #2, we will be exploring nonparametric methods of slope.

II. Summary

First we will be taking a look at the head of our dataset. This will provide us with information to answer question #1, showing the types of variables that we will be analyzing. We firstly begin by removing the observations where fertility has missing values.

Head of data after removing missing values:

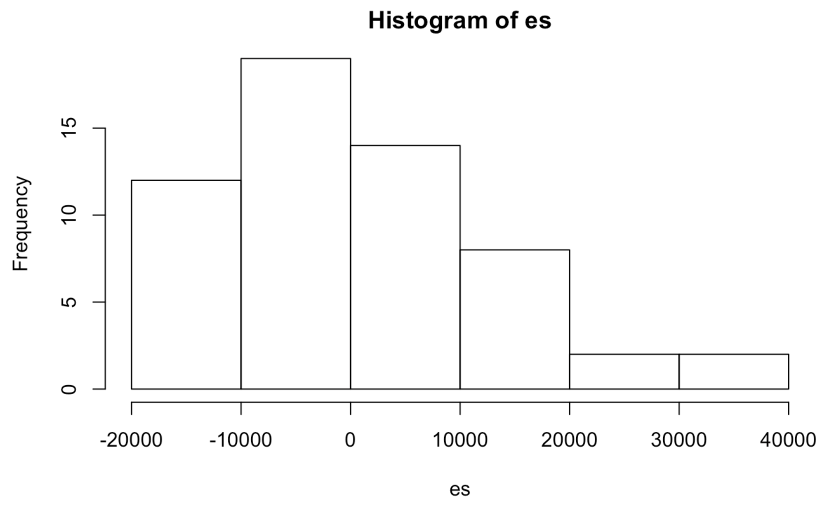
fertility	GDP
<dbl>	<int>
1.28	31187
2.10	853
2.29	14462
1.66	29257
1.67	51991
2.21	2700

Here, we see that our data contains two columns, the first named fertility and the second named GDP. From the head of the data, we can see that the two variables are numerical. Since we are trying to learn about the relationship between the two variables, we can treat these two variables as bivariate data to help us learn if there is any association between them. We must also remember that we removed the rows with missing values. Therefore, it is potentially better to get a confidence interval. This leads us into the use of Bootstrap confidence intervals for the slope.

III. Analysis

Firstly we assume a simple linear regression model in which B1 is the slope and multiplied by fertility rate and Y is GDP. Finding this will tell us about the relationship between GDP and fertility. In this case, we are using the errors to bootstrap.

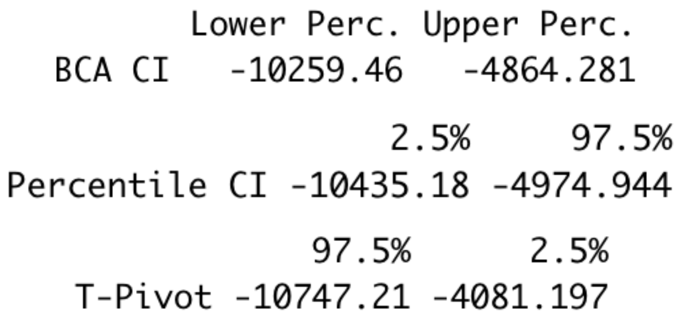
Distribution of bootstrapped errors:



From the histogram of the errors, we can see that the eros are skewed to the right.

Now we are ready to find our confidence intervals of the slope. Here we will be finding a 95% confidence interval.

Confidence intervals for the slope using BCA, percentile, and T-pivot.



```
         Lower Perc. Upper Perc.
BCA CI   -10259.46   -4864.281

                   2.5%     97.5%
Percentile CI -10435.18 -4974.944

             97.5%      2.5%
T-Pivot -10747.21 -4081.197
```

Our BCA confidence interval of the slope is (-10,029.46, -4864.281). The percentile confidence interval is (-10,435.18, -4974.994) and our t-pivot CI is (-10747.21, -4081.197).

IV. Interpretation

Based on the histogram, we see that our errors are skewed to the right. The percentile and BCA confidence intervals are a little different indicating that there is some skewness or bias in our estimator. Judging from the skewed distribution of eros it is likely to expect these results. However since the errors are skewed, we are more inclined to use BCA as a more accurate CI, because it accounts for bias. Therefore, we are 95% confident that for every live birth born per woman of a certain age per year (the rate), then we can expect the GDP of the country to be less by between -10,029.46 and -4864.281.

V. Conclusion

This analysis was conducted to explore the relationship between GDP and fertility rates. Through bootstrapping for the slope, we uncovered that there was a negative relationship between GDP and Fertility. Based on the bootstrap distribution of the errors, we used BCA to discover that we are 95% confident that for every live birth born per woman of a certain age per year, the GDP decreases by between -10,029.46 and -4864.281. However, these findings were made by using bootstrapping, so it should be noted that the distribution that bootstrapping uses is not entirely robust.

Contributions: Miguel Morales both Topic 1 and Topic 2

Code Appendix

Topic 1

```
some.numbers = c(1,4,6,6,7,25,13,13,12,14,16,14)
some.groups = rep(c("A","B"),times = c(6,6))
dat.data = data.frame(numbers = some.numbers,groups = some.groups)
dat.data
all.perms = sapply(1:3000, function(i){
  the.numbers <- dat.data$numbers
  the.groups <- dat.data$groups
  new.labels <- sample(the.groups,length(the.groups),replace = FALSE) # shuffles groups
  group.1.var = var(the.numbers[new.labels == levels(the.groups)[1]]) # finds var for group 1
  group.2.var = var(the.numbers[new.labels == levels(the.groups)[2]]) # finds var for group 2
  difference.in.vars= group.1.var-group.2.var #finds difference in var
  return(difference.in.vars)
})
head(all.perms)
hist(all.perms)
sample.vars = aggregate(numbers ~ groups,dat.data,var)[,2] # finds var per group
sample.vars
difference = sample.vars[1] - sample.vars[2]
difference
p.value.two = mean(abs(all.perms) >= abs(difference)) #calculates two-sided p-value
p.value.less = mean(all.perms <= difference) #calculates p-value for "less than" alternative
p.value.greater = mean(all.perms >= difference) #calculates p-value for "greater than" alternative
p.value.two<- rbind(p.value.two)
rownames(p.value.two)<-c("p-value =")
p.value.two
sample.vars<-rbind(sample.vars)
rownames(sample.vars)<- c("Variance of both groups")
colnames(sample.vars)<- c("Group 1", "Group 2")
sample.vars

difference<-rbind(difference)
rownames(difference)<- c("Difference in Variance")
difference
```

Topic 2 Part 1

```
library(dplyr)
crowdedness<-read.csv("~/Downloads/crowdedness.csv")
```

```
crowdedness
crowdedness <-subset(crowdedness, select=c("Crowdedness","GDP"))
colSums(is.na(crowdedness))

crowdedness<- na.omit(crowdedness)
crowdedness

plot(Crowdedness~GDP, crowdedness, main= "Crowdedness vs GDP")

library(KernSmooth)

makeKernelSmoother <- function(bw, mydata){
  fit.kernel <- locpoly(mydata$GDP,mydata$Crowdedness, degree = 0, bandwidth = bw)
  #range.x = range(crowdedness$GDP)
  return(fit.kernel)
}

estPhiHat <- function(x, fit.kernel){
  ind.closest.x <- which.min(abs(fit.kernel$x-x))
  fit.kernel$y[ind.closest.x]
}

bws <- seq(2000, 10000, by = 2000)

plot(Crowdedness~GDP,crowdedness,  main= "Kernel smoothers on Crowdedness~GDP",col =
rgb(0,0,0, .5))

i<-1
for(mybws in bws){
  kern<-makeKernelSmoother(mybws, crowdedness)
  lines(kern, type = "l", col = i)
  i<-i+1
}
legend("topright", legend = bws, col = 1:6, lty = 1)
calcTestMSPE <- function(testdata, fit_train){
  yhats_test <- sapply(testdata$GDP, estPhiHat, fit.kernel = fit_train)
  MSPEs<-mean((testdata$GDP - yhats_test)^2)
  return(MSPEs)
}
```

```
n <- nrow(crowdedness)
k <- 10

set.seed(77)

folds <- cut(1:n, breaks = k, labels = FALSE)
folds <- sample(folds)

kCV_MSPE.1 <- rep(NA,k)
kCV_MSPE.2 <- rep(NA,k)
kCV_MSPE.3 <- rep(NA,k)
kCV_MSPE.4 <- rep(NA,k)
kCV_MSPE.5 <- rep(NA,k)

for(current_k in 1:k){

  test <- crowdedness[folds==current_k, ]
  train <- crowdedness[folds!=current_k, ]

  fit_train_1 <- makeKernelSmoother(2000, train)
  fit_train_2 <- makeKernelSmoother(4000, train)
  fit_train_3 <- makeKernelSmoother(6000, train)
  fit_train_4 <- makeKernelSmoother(8000, train)
  fit_train_5 <- makeKernelSmoother(10000, train)

  MSPE.1 <- calcTestMSPE(testdata =test,fit_train_1)
  MSPE.2 <- calcTestMSPE(testdata =test,fit_train_2)
  MSPE.3 <- calcTestMSPE(testdata =test,fit_train_3)
  MSPE.4 <- calcTestMSPE(testdata =test,fit_train_4)
  MSPE.5 <- calcTestMSPE(testdata =test,fit_train_5)

  kCV_MSPE.1[current_k] <- MSPE.1
  kCV_MSPE.2[current_k] <- MSPE.2
  kCV_MSPE.3[current_k] <- MSPE.3
  kCV_MSPE.4[current_k] <- MSPE.4
  kCV_MSPE.5[current_k] <- MSPE.5
}

mean(kCV_MSPE.1)
mean(kCV_MSPE.2)
mean(kCV_MSPE.3)
mean(kCV_MSPE.4)
mean(kCV_MSPE.5)

foldmspes<-rbind(mean(kCV_MSPE.1),mean(kCV_MSPE.2),mean(kCV_MSPE.3),mean(kCV_
MSPE.4),mean(kCV_MSPE.5))
colnames(foldmspes)<- c("10-fold mspe")
rownames(foldmspes) <-c("bw 2000:", "bw 4000:", "bw 6000:", "bw 8000:","bw 10000:")
foldmspes

bw5<- makeKernelSmoother(10000, crowdedness)
est878<-estPhiHat(878,bw5)
est5945<-estPhiHat(5945,bw5)
estimations1<-rbind(est878,est5945)
rownames(estimations1)<-c(" at GDP = 878", " at GDP = 5945")
colnames(estimations1)<-c("Crowdedness Estimation")
estimations1

s= k/n
s

fit.loess1 <- loess(Crowdedness ~ GDP, crowdedness, degree = 1,span = .3)
fit.loess2 <- loess(Crowdedness ~ GDP, crowdedness, degree = 1,span = .4)
fit.loess3 <- loess(Crowdedness ~ GDP, crowdedness, degree = 1,span = .5)
fit.loess4 <- loess(Crowdedness ~ GDP, crowdedness, degree = 1,span = .6)
fit.loess5 <- loess(Crowdedness ~ GDP, crowdedness, degree = 1,span = .7)

xrange <- range(crowdedness$GDP)
xgrid <- seq(xrange[1], xrange[2], length = 100)
yhats.loess1 <- predict(fit.loess1, data.frame(GDP = xgrid))
yhats.loess2 <- predict(fit.loess2, data.frame(GDP = xgrid))
yhats.loess3 <- predict(fit.loess3, data.frame(GDP = xgrid))
yhats.loess4 <- predict(fit.loess4, data.frame(GDP = xgrid))
yhats.loess5 <- predict(fit.loess5, data.frame(GDP = xgrid))

bws<-seq(1,5, by =1)
plot(Crowdedness ~ GDP, crowdedness, main = "Loess curve")
lines(xgrid, yhats.loess1, col = 1)
lines(xgrid, yhats.loess2, col = 2)
lines(xgrid, yhats.loess3, col = 3)
lines(xgrid, yhats.loess4, col = 4)
lines(xgrid, yhats.loess5, col = 5)
legend("topright", legend = bws, col = 1:6, lty = 1)

# k-fold subsetting
n <- nrow(crowdedness)
n.train <- round(n*.7)
n.test <- n - n.train

inds.train <- sample(1:n, size = n.train)
inds.test <- (1:n)[-inds.train]

k <- 10
folds <- cut(1:n, breaks = k, labels = FALSE)

crowdedness.train <- crowdedness[inds.train,]
crowdedness.test <- crowdedness[inds.test,]

yhats_test1 <- predict(fit.loess1, crowdedness.test)
yhats_test2 <- predict(fit.loess2, crowdedness.test)
yhats_test3 <- predict(fit.loess3, crowdedness.test)
yhats_test4 <- predict(fit.loess4, crowdedness.test)
yhats_test5 <- predict(fit.loess5, crowdedness.test)

loess1<-mean((crowdedness.test$Crowdedness - yhats_test1)^2)
loess2<-mean((crowdedness.test$Crowdedness - yhats_test2)^2)
loess3<mean((crowdedness.test$Crowdedness - yhats_test3)^2)
loess4<-mean((crowdedness.test$Crowdedness - yhats_test4)^2)
loess5<-mean((crowdedness.test$Crowdedness - yhats_test5)^2)

loessmspe<-rbind(loess1,loess2, loess3, loess4, loess5)
rownames(loessmspe)<- c("s = .3","s = .4","s = .5","s = .6","s = .7")
colnames(loessmspe)<- c("MSPE")
loessmspe

loess878<-predict(fit.loess4, newdata = data.frame(GDP = 878))
loess5945<-predict(fit.loess4, newdata = data.frame(GDP = 5945))
```

~~~
rbinds<- rbind(loess878,loess5945)
rbinds

Part 2
```{r}
crowdedness<-read.csv("~/Downloads/crowdedness.csv")

crowdedness <-subset(crowdedness, select=c("fertility","GDP"))

colSums(is.na(crowdedness))

crowdedness<- na.omit(crowdedness)
head(crowdedness)

boxplot(crowdedness$fertility,main="Boxplot fertility")
```

```{r}
set.seed(77)
fit.mlr <- lm(DH ~ MH + FH, daughters)
es.mlr <- fit.mlr$residuals
hist(es.mlr)
```



```{r}
findTheta <- function(x,y){
  cor(x,y)
}
rho.obs <- findTheta(crowdedness$GDP, crowdedness$fertility)

# Obtain bootstrapped rho*'s
B <- 4000
n <- nrow(crowdedness)

set.seed(77)
boot.thetas <- rep(NA, B)
for(b in 1:B){
  boot.inds <- sample(1:n, replace = TRUE)
  boot.sample <- crowdedness[boot.inds,]
~~~

```
  boot.thetas[b] <- findTheta(boot.sample$GDP, boot.sample$fertility)
}

hist(boot.thetas, main = "Histogram of bootstrapped rho")
```

```{r}
# percentile rho
percentileCI<- rbind(quantile(boot.thetas, c(0.025, .975)))
rownames(percentileCI)<- c("Percentile CI")
percentileCI
```

```{r}
# bca rho
alpha <- 0.05
po<-mean(boot.thetas<=rho.obs)
Z0<-qnorm(po)
Za<-qnorm(1-alpha/2)

leave.one.out.theta = sapply(1:nrow(crowdedness),function(i){
  leave.out.data = crowdedness[-i,c('GDP','fertility')]
  cor(leave.out.data$GDP,leave.out.data$fertility)
})
theta.minus.one = mean(leave.one.out.theta)
a = sum( (theta.minus.one - leave.one.out.theta)^3)/( 6 *(sum( (theta.minus.one -
leave.one.out.theta)^2))^(3/2) )
Zu = (Z0+Za)/(1-a*(Z0+Za)) + Z0 #upper limit for Z
Zl = (Z0-Za)/(1-a*(Z0-Za)) + Z0 #Lower limit for Z
lower.percentile = pnorm(Zl,lower.tail = TRUE) #percentile for Z
upper.percentile = pnorm(Zu,lower.tail = TRUE) #percentile for Z
ci.bca = rbind(as.numeric(quantile(boot.thetas,c(lower.percentile,upper.percentile))))
rownames(ci.bca)<-c("BCA CI")
colnames(ci.bca)<-c("Lower", "Upper")
ci.bca

rbind(percentileCI,ci.bca)

# either work because distribution of rho is equal
```

```
set.seed(77)
fit.slr <- lm(GDP ~ fertility, crowdedness)
beta1_hat <- fit.slr$coefficients["fertility"]
beta1_hat
es <- fit.slr$residuals
hist(es)

yhats <- fit.slr$fitted.values

# Example of percentile 95% CI SLOPE:

B =10000
findTheta <- function(mydata){
  fit.slr <- lm(GDP ~ fertility, mydata)
  beta1_hat <- fit.slr$coefficients["fertility"]
  return(beta1_hat)
}

boot.thetas <- rep(NA, B)
for(b in 1:B){
  boot.es <- sample(es, replace = TRUE)
  boot.ys <- yhats + boot.es
  boot.sample <- data.frame(fertility = crowdedness$fertility, GDP = boot.ys)
  boot.thetas[b] <- findTheta(boot.sample)
}

quantile(boot.thetas, c(0.025, .975))

model <- lm(GDP~fertility,crowdedness)
boot_reps1<-replicate(10000,{
  data_boot<-crowdedness[sample(nrow(crowdedness),replace=TRUE),c('fertility','GDP')]
  model<-lm(GDP ~ fertility, data_boot)
  return(coefficients(model)['fertility'])
})

po = mean(boot_reps1 <= coefficients(model)['fertility'])
Z0 = qnorm(po)
Za = qnorm(1-alpha/2)

### Estimating a:
```

```
leave.one.out.theta1 = sapply(1:nrow(crowdedness),function(i){
  leave.out.data = crowdedness[-i,c('fertility','GDP')]
  model<-lm(GDP~fertility, leave.out.data)
  return(coefficients(model)['fertility'])
})

theta.minus.one = mean(leave.one.out.theta1)
a = sum( (theta.minus.one - leave.one.out.theta1)^3)/( 6 *(sum( (theta.minus.one -
leave.one.out.theta1)^2))^(3/2) )

Zu = (Z0+Za)/(1-a*(Z0+Za)) + Z0 #upper limit for Z
Zl = (Z0-Za)/(1-a*(Z0-Za)) + Z0 #Lower limit for Z
lower.percentile = pnorm(Zl,lower.tail = TRUE) #percentile for Z
upper.percentile = pnorm(Zu,lower.tail = TRUE) #percentile for Z

oci.bca = as.numeric(quantile(boot_reps1,c(lower.percentile,upper.percentile)))
rbind(oci.bca)

oci.bca<-rbind(oci.bca)
rownames(oci.bca)<-c("BCA CI")
colnames(oci.bca)<-c("Lower Perc.", "Upper Perc.")
oci.bca

thperc<-quantile(boot_reps1, c(0.025, .975))
theperc<-rbind(thperc)
rownames(theperc)<- c("Percentile CI")
theperc
#Bootstrap t-pivot method for slope
se_beta1hat <- summary(fit.slr)$coefficients[2,2]

boot.ts <- rep(NA, B)
for(b in 1:B){
  boot.es <- sample(es, replace = TRUE)
  fit.boot <- lm(boot.es ~ crowdedness$fertility)
  boot.ts[b] <- summary(fit.boot)$coefficients[2,3]
}

tquants <- quantile(boot.ts, c(0.975, .025))
tslope<-beta1_hat - tquants*se_beta1hat
tslope
```

~~~
tslope<-rbind(tslope)
rownames(tslope)<-c("T-Pivot")
tslope
```
~~~