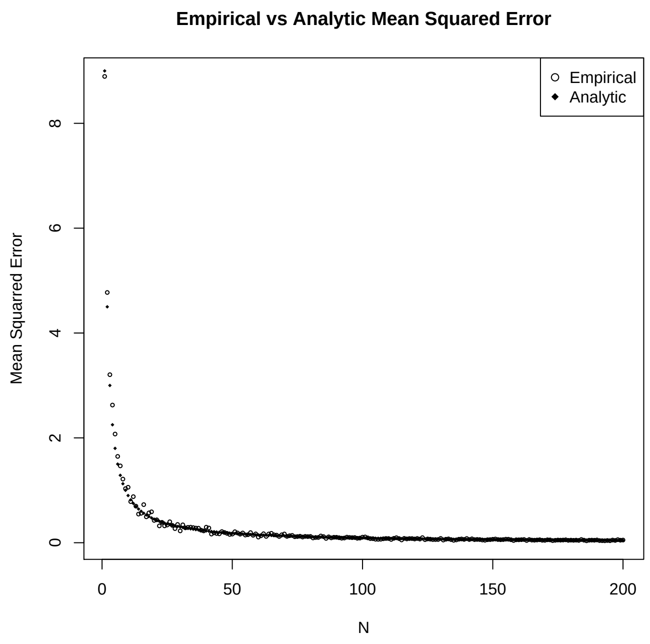
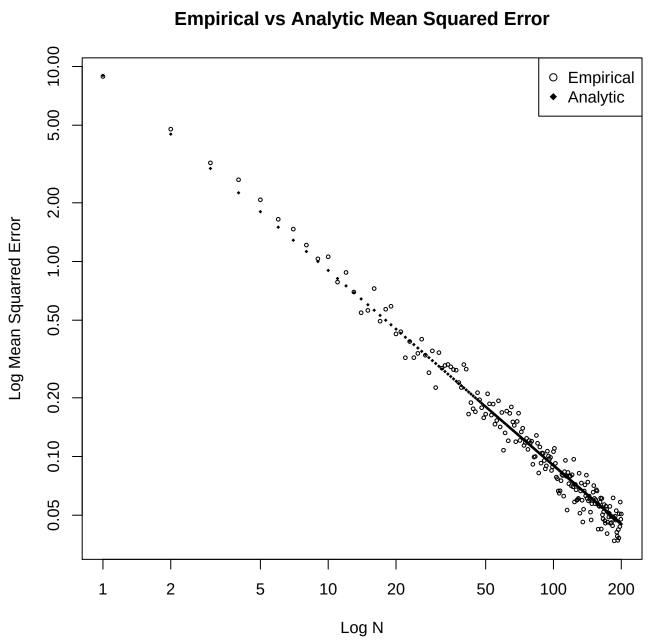
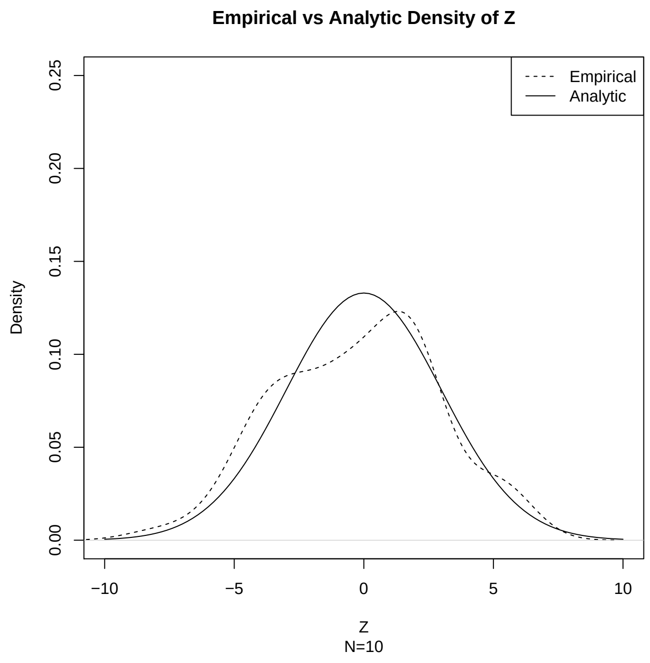
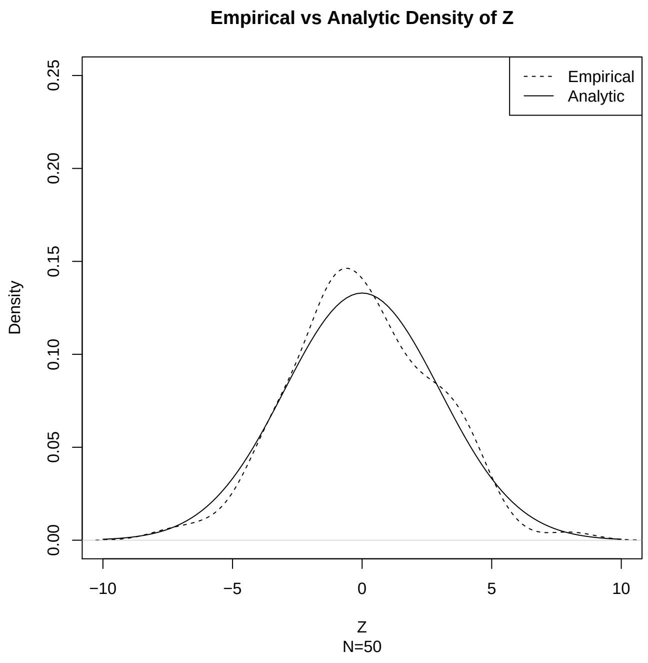
1. a) I've only included the graphs here, the commands used to generate them can be found in the R supplement in the back. For all of these, B, the number of trials for a particular N, was set to 100.

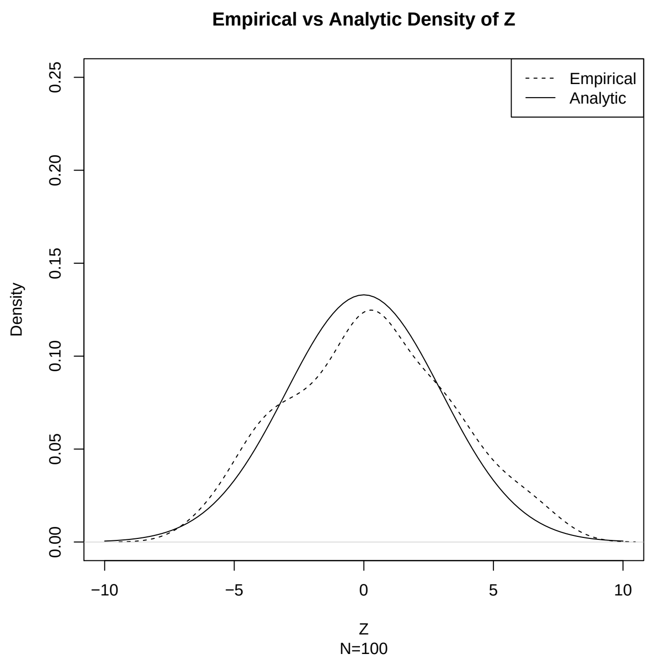




b) I chose N to be 10, 50, and 100. For each of these, the density plot is again off of $B = 100$ trials.







2. a) For $\sigma = 1$

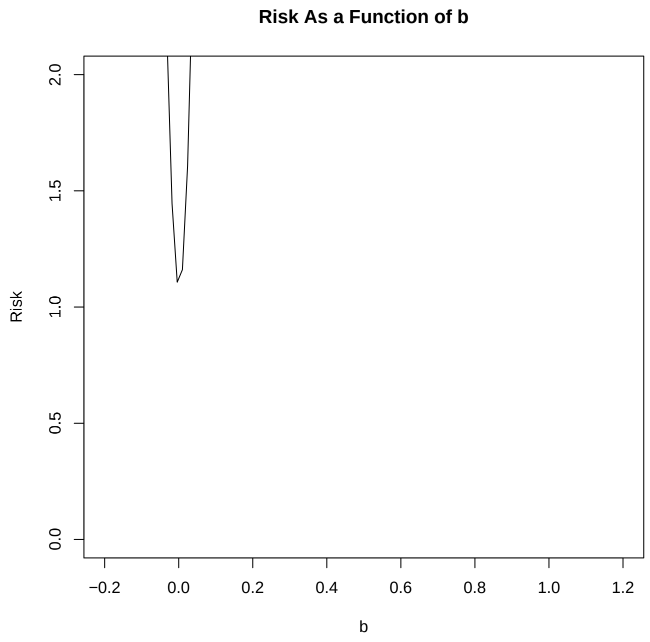
The risk of the MLE estimator is just the variance. Thus,

$$R(Z, \theta) = \sum_{i=1}^{1000} \sigma^2 = \sum_{i=1}^{1000} 1 = 1000$$

The risk of a shrinkage estimator will be:

$$R(\hat{\theta}, \theta) = b^2 n\sigma^2 + (1-b)^2\|\theta\|^2 = b^2 1000 + (1-b)^2 \sum_{i=1}^{1000} \frac{1}{i^4} = b^2 1000 + (1-b)^2 * 1.082323$$

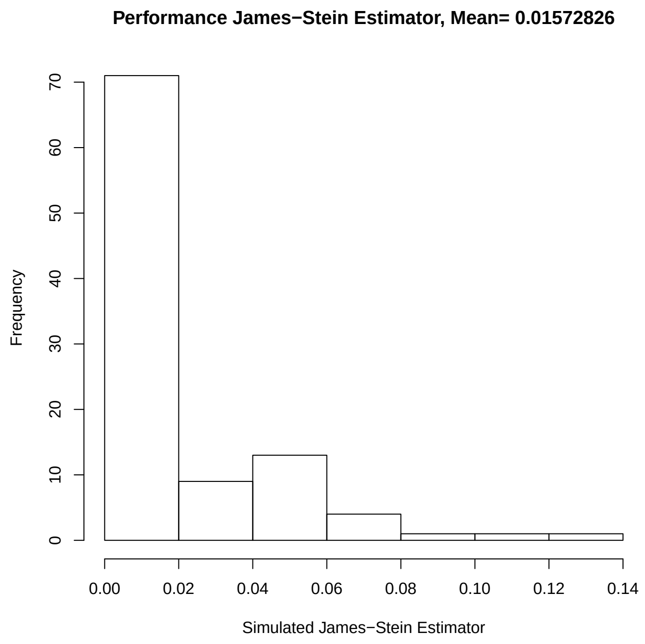
Which is plotted here:



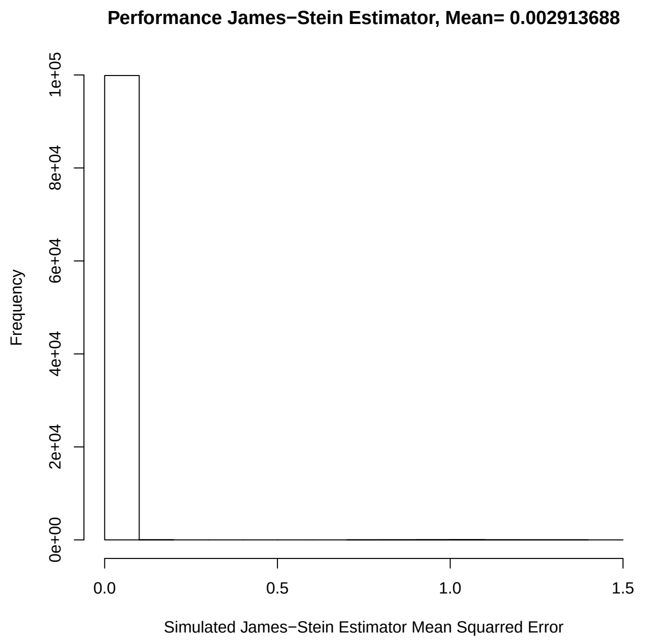
As the graph suggests, the minimum is achieved at

$$b_* = \frac{\|\theta\|^2}{n\sigma^2 + \|\theta\|^2} = \frac{1.082323}{1 + 1.082323} = .0010812$$

The James-Stein estimator seems to be a good approximation of this; it appears to be centered at the optimal b (the mean is higher because negative values are excluded), with some random variance:



Setting c^2 to 1.082323, we get a Pinsker Bound of $\frac{1*1.082323}{1+1.082323} = .52976$. The MLE risk blows right by this by a factor of 2000, partly because $\sigma_n^2 \neq \frac{\sigma^2}{n}$. The simulated mean squared error of the JS estimator is, as it should be, always coming in much below this threshold"



b) For $\sigma = \frac{1}{n}$

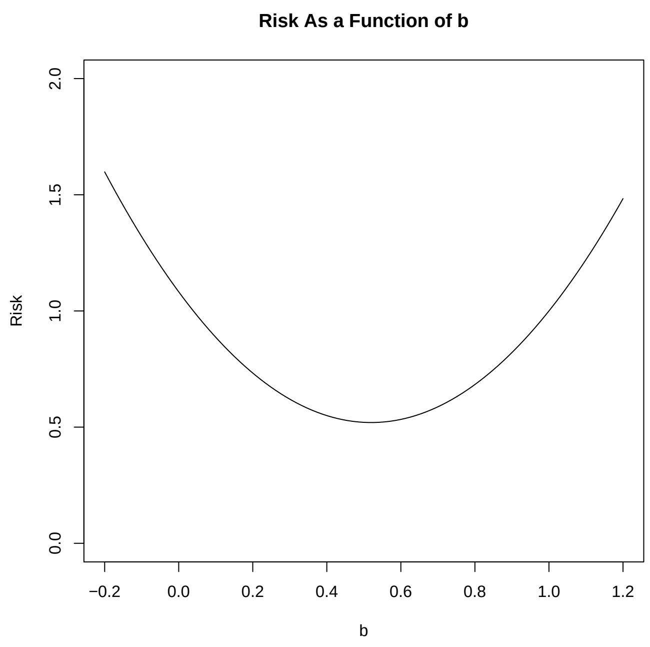
The risk of the MLE estimator is just the variance. Thus,

$$R(Z, \theta) = \sum_{i=1}^{1000} \sigma^2 = \sum_{i=1}^{1000} \frac{1}{1000} = 1$$

The risk of a shrinkage estimator will be:

$$R(\hat{\theta},\theta) = b^2 n\sigma^2 + (1-b)^2\|\theta\|^2 = b^2 + (1-b)^2 \sum_{i=1}^{1000} \frac{1}{i^4} = b^2 + (1-b)^2 * 1.082323$$

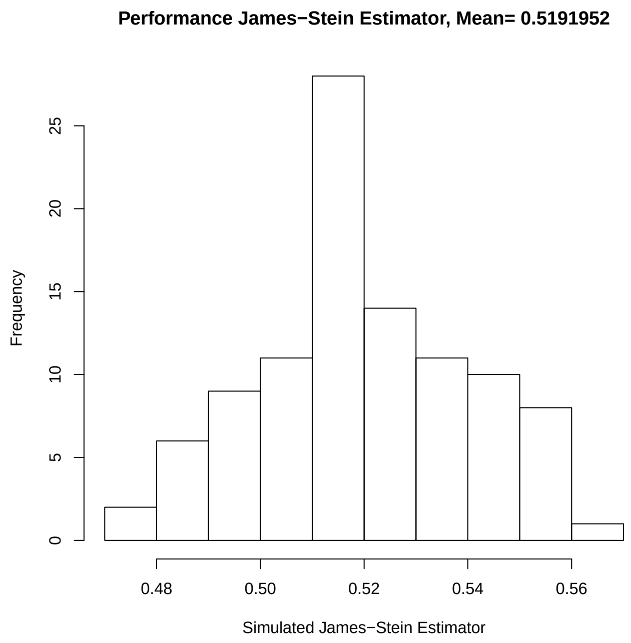
Which is plotted here:



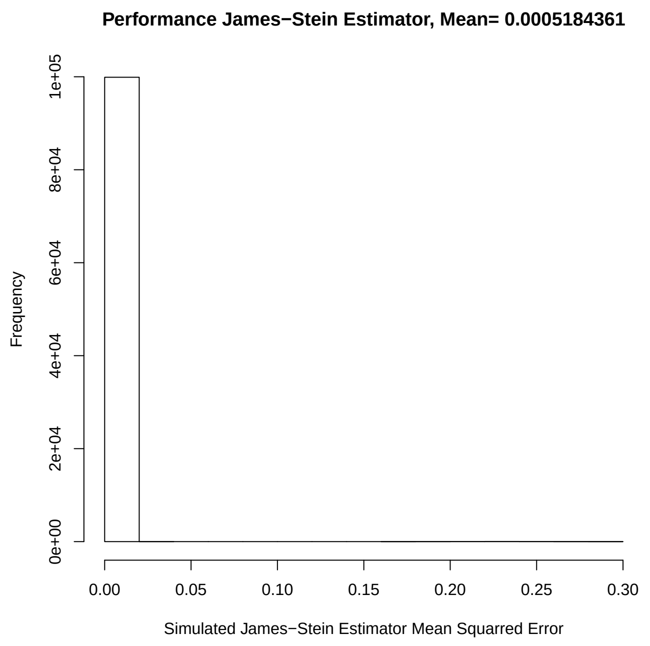
As the graph suggests, the minimum is achieved at

$$b_* = \frac{\|\theta\|^2}{n\sigma^2 + \|\theta\|^2} = \frac{1.082323}{1 + 1.082323} = .5197671$$

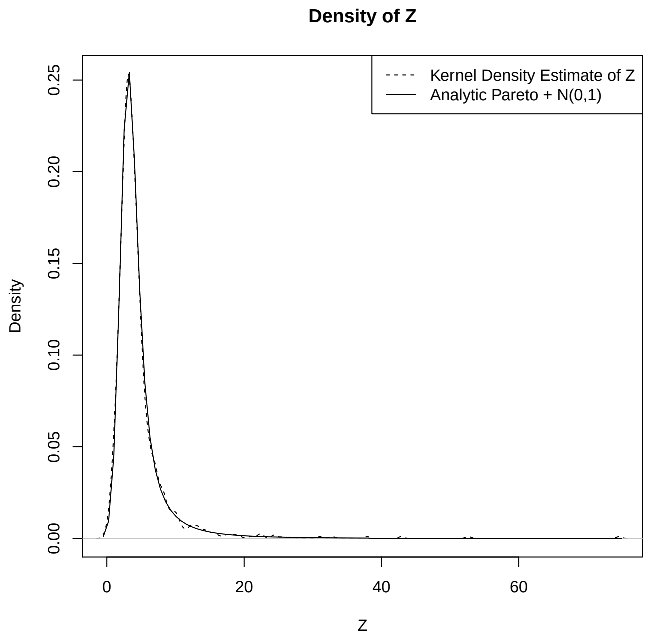
The James-Stein estimator seems to be a good approximation of this; it appears to be centered at the optimal b, with some random variance:

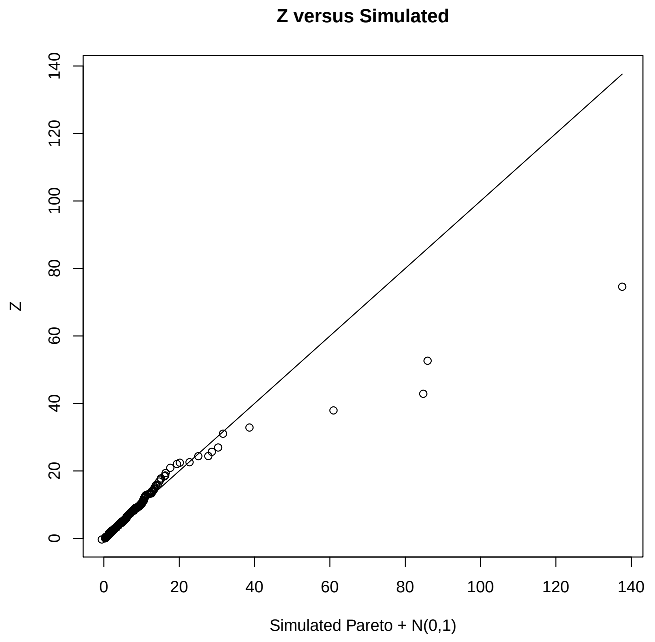


With the same Pinkser Bound, with the variance assumption actually holding, the MLE risk is still about twice as high at 1. The simulated mean squared error of the JS estimator comes in far below, as one would expect.



3. After some investigation, it is my belief that the actual θ vector was generated by a Pareto distribution, or a distribution that is quite similar to it. Choosing a shape parameter of 2 and a minimum of 2.4 for the Pareto distribution, I certainly seem to be very close:





This is about as good a fit as you will see with a parametric model. Its hard to judge the fit of the tail, but even if it is off, since the signal overpowers the noise there, it doesn't matter that much for this exercise.

Armed with the assumption that $\theta \sim Pareto(2, 2.4)$, I can calculate the density function of θ conditioned on the particular Z:

$$\begin{aligned} f_\theta(\theta \,|\, Z) &= f_\theta(\theta \,|\, \theta + N(0,1) = z) \\ &= \frac{2 \cdot 2.4^2 \theta^{-3}}{\int_{2.4}^{\infty} 2 \cdot 2.4^2 x^{-3} \phi(z-x) dx} \end{aligned}$$

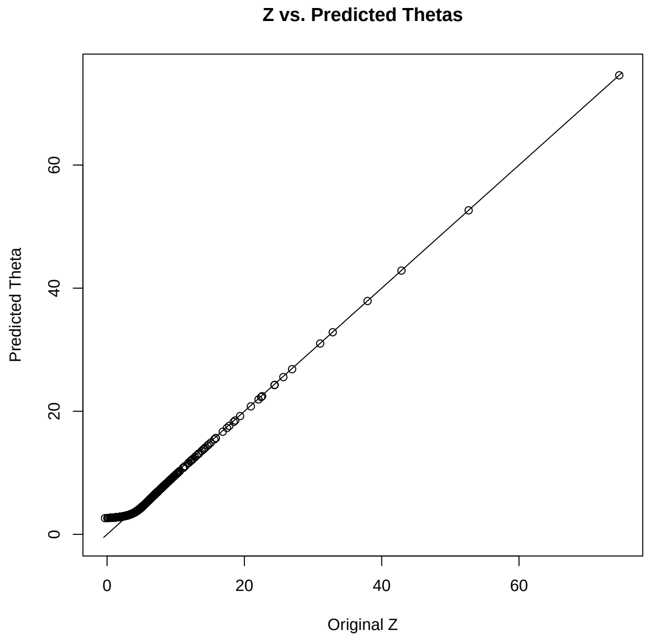
Using this, I can calculate the expected mean square error for a particular estimate $\hat{\theta}_i$ conditioned on Z_i:

$$\mathrm{E}[(\theta_i - \hat{\theta}_i)^2 \mid Z_i] = \int_{2.4}^{\infty} (\theta_i - \hat{\theta}_i)^2 f_\theta(\theta \mid Z_i) d\theta$$

Combining all of this, I chose the $\hat{\theta}_i$ which minimized this expected mean squared error:

$$\hat{\theta}_i = \operatorname*{argmin}_b \mathrm{E}[(\theta_i - b)^2 \mid Z_i]$$

This was all implemented numerically in R. The one caveat is that the numerical integral ran into precision issues in calculation when Z was large. I assume the reason is the integral was summing over values smaller than can be represented in double precision. Thus, for Z larger than 30, I instead used $\hat{\theta}_i = \frac{Z_i - 1}{Z_i}$, which is what the James Stein estimate would be for that observation by itself. The end result is this:



Values of Z below 2.4 all got $\hat{\theta}_i$ higher than 2.4 (since this was the assumed minimum), and there was a general pull towards 3 or so, with larger values of Z getting shrunk less than smaller values greater than 3.

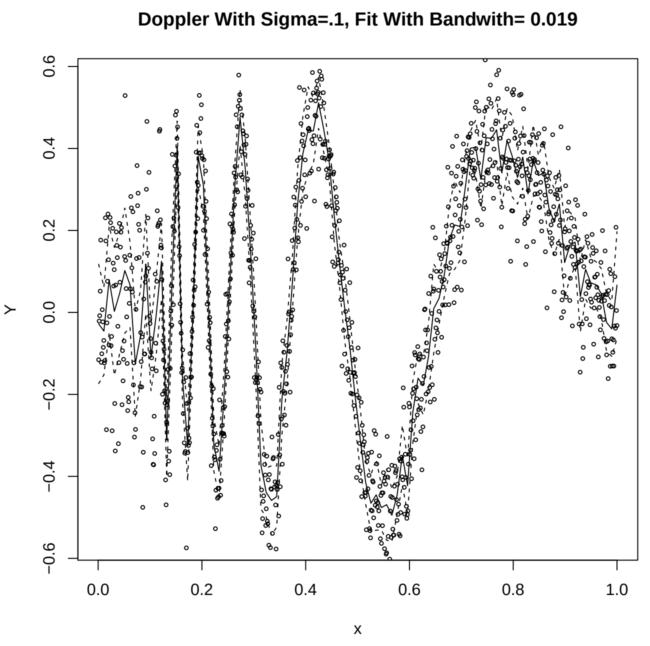
4. We will begin by calculating the risk for a given $\bar{X}_n$

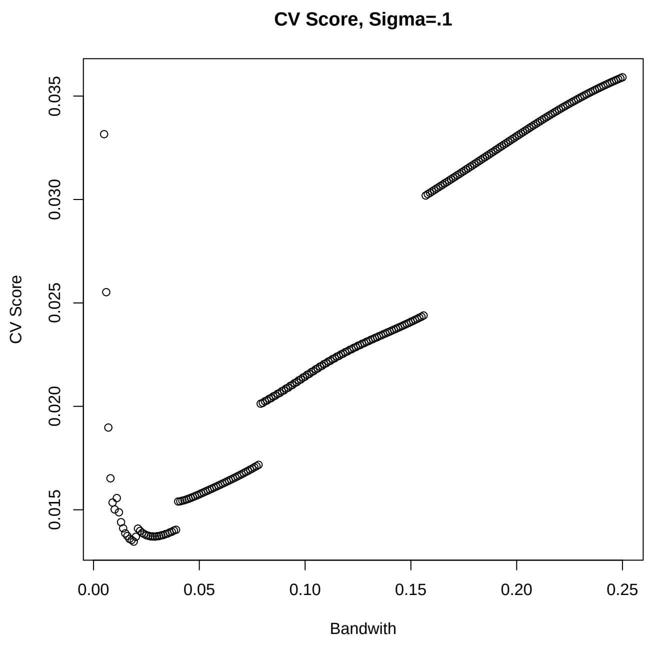
$$\begin{aligned}
\mathrm{E}\left[\left(f_\theta(x)-f_{\bar X_n}(x)\right)^2 \mid \bar X_n\right] &= \int_{-\infty}^{\infty}(f_\theta(x)-f_{\bar X_n}(x))^2dx\\
&= \int_{-\infty}^{\infty} f_\theta(x)^2dx+\int_{-\infty}^{\infty} f_{\bar X_n}(x)^2dx-2\int_{-\infty}^{\infty} f_\theta(x)f_{\bar X_n}(x)dx\\
&= 2\int_{-\infty}^{\infty}\left(\frac{1}{\sqrt{2\pi}}e^{\frac{-(x-\mu)^2}{2}}\right)^2dx-2\int_{-\infty}^{\infty}\left(\frac{1}{\sqrt{2\pi}}e^{\frac{-(x-\theta)^2}{2}}\right)\left(\frac{1}{\sqrt{2\pi}}e^{\frac{(-(x-\bar X_n)^2}{2}}\right)dx\\
&= \frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty}\frac{1}{\sqrt{\pi}}e^{-(x-\mu)^2}dx-\frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty}\frac{1}{\sqrt{\pi}}e^{\frac{-(x-\theta)^2-(x-\bar X_n)^2}{2}}dx\\
&= \frac{1}{\sqrt{\pi}}-\frac{e^{-\frac14(\theta-\bar X_n)^2}}{\sqrt{\pi}}\int_{-\infty}^{\infty}\frac{1}{\sqrt{\pi}}e^{-x^2+x\theta+x\bar X_n-\frac{\theta^2+X_n^2}{2}+\frac{\theta^2-2\theta\bar X_n+X_n^2}{4}}dx\\
&= \frac{1}{\sqrt{\pi}}-\frac{e^{-\frac14(\theta-\bar X_n)^2}}{\sqrt{\pi}}\int_{-\infty}^{\infty}\frac{1}{\sqrt{\pi}}e^{-x^2+x\theta+x\bar X_n-\frac{\theta^2+2\theta\bar X_n+\bar X_n^2}{4}}dx\\
&= \frac{1}{\sqrt{\pi}}-\frac{e^{-\frac14(\theta-\bar X_n)^2}}{\sqrt{\pi}}\int_{-\infty}^{\infty}\frac{1}{\sqrt{\pi}}e^{-(x-\frac{\theta+X_n}{2})^2}dx\\
&= \frac{1}{\sqrt{\pi}}-\frac{e^{-\frac14(\theta-\bar X_n)^2}}{\sqrt{\pi}}
\end{aligned}$$

Now, taking the expectation of this over all $\bar X_n$ will give us the unconditional expectation of the risk.

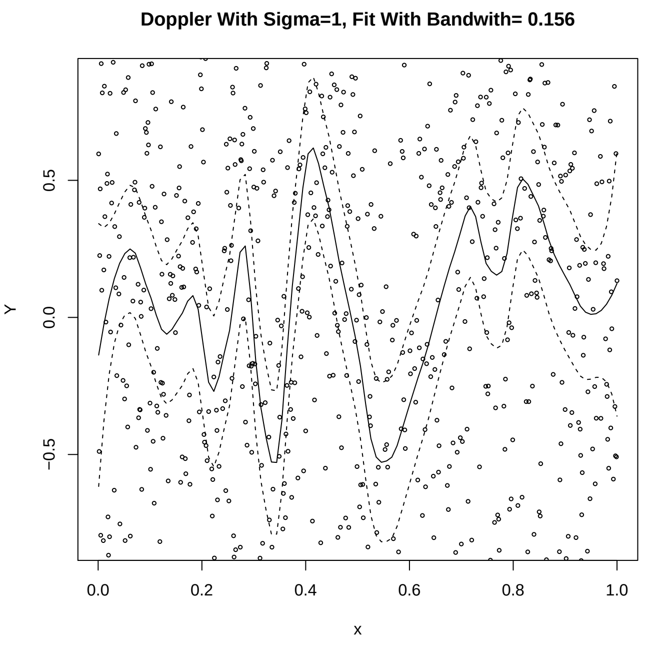
$$\begin{aligned}
\mathrm{E}\left[\left(f_\theta(x)-f_{\bar X_n}(x)\right)^2\right] &= \mathrm{E}\left[\mathrm{E}\left[\left(f_\theta(x)-f_{\bar X_n}(x)\right)^2 \mid \bar X_n\right]\right]\\
&= \int_{-\infty}^{\infty}\left(\frac{1}{\sqrt{\pi}}-\frac{e^{-\frac14(\theta-\bar X_n)^2}}{\sqrt{\pi}}\right)\left(\frac{\sqrt{n}}{\sqrt{2\pi}}e^{\frac{-n(\theta-\bar X_n)^2}{2}}\right)d\bar X_n\\
&= \frac{1}{\sqrt{\pi}}-\frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty}\frac{\sqrt{n}}{\sqrt{2\pi}}e^{\frac{-(\frac12+n)(\theta-\bar X_n)^2}{2}}d\bar X_n\\
&= \frac{1}{\sqrt{\pi}}-\frac{1}{\sqrt{\pi}}\sqrt{\frac{n}{\frac12+n}}\int_{-\infty}^{\infty}\frac{\sqrt{\frac12+n}}{\sqrt{2\pi}}e^{\frac{-(\frac12+n)(\theta-\bar X_n)^2}{2}}d\bar X_n\\
&= \frac{1}{\sqrt{\pi}}\left(1-\sqrt{\frac{2n}{1+2n}}\right)
\end{aligned}$$

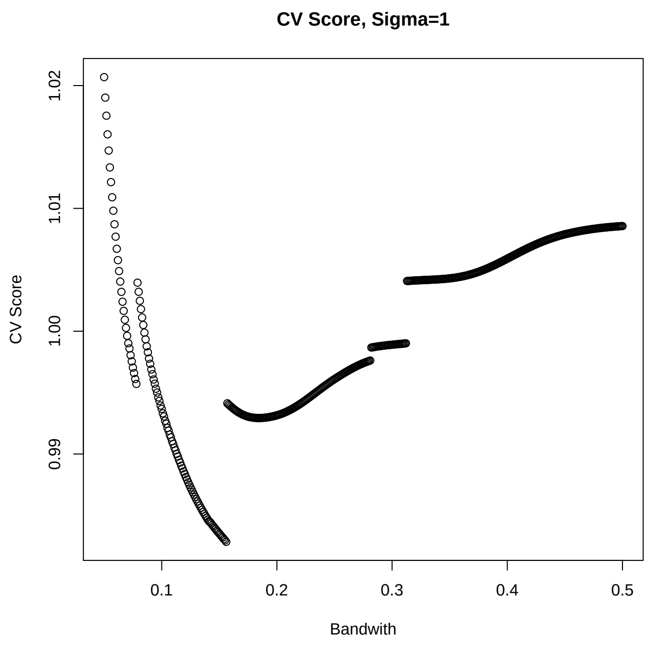
5. Doppler function with $\sigma = .1$:





Doppler function with $\sigma = 1$:





6. Per the definition of leave-one-out cross-validation, we have that

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{r}_{(-i)}(x)\right)^2$$

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \sum_{j\neq i} Y_i l_{j,(-i)}(x)\right)^2$$

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \frac{1}{\sum_{j\neq i} l_j(x)}\sum_{j\neq i} Y_i l_j(x)\right)^2$$

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \frac{\hat{r}_n(x) - Y_i l_i(x)}{1 - l_i(x)}\right)^2$$

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - Y_i l_i(x) - \hat{r}_n(x) + Y_i l_i(x)}{1 - l_i(x)}\right)^2$$

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{r}_n(x)}{1 - l_i(x)}\right)^2$$

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{r}_n(x)}{1 - L_{ii}}\right)^2$$

This is the R code I used for this homework:

```
library('parallel','locfit')

# Option on which section of code to run

##################################################
# Problem 1
##################################################

#Part a
m<-1
s<-3
b<-100
nseq<-1:200
uhat<-function(n) mean(rnorm(n,m,s))
meansqerr<-function(n) mean((m-sapply(rep(n,times=b),uhat))^2)
empmsqerr<-sapply(nseq, meansqerr)
anamsqerr<-sapply(nseq, function(n) (s^2)/n)

pdf('hw1_1_a_reg.pdf')
plot(1:200,empmsqerr,main="Empirical vs Analytic Mean Squared Error", pch=1, cex=.5, xla
points(1:200,anamsqerr,cex=.5, pch=18)
legend('topright', legend=c('Empirical','Analytic'), pch=c(1,18))
dev.off()

pdf('hw1_1_a_log.pdf')
plot(1:200,empmsqerr,main="Empirical vs Analytic Mean Squared Error", pch=1, cex=.5, xla
points(1:200,anamsqerr,cex=.5, pch=18)
legend('topright', legend=c('Empirical','Analytic'), pch=c(1,18))
```

```
dev.off()

#Part b
nseq<-c(10,50,100)
genZ<-function(n) sqrt(n)*(m-sapply(rep(n,times=b),uhat))
for (n in nseq) {
    pdf(paste('hw1_1_b_',toString(n),'.pdf', sep=""))
    plot(density(genZ(n)), main="Empirical vs Analytic Density of Z", sub=paste('N=',toS
    curve(dnorm(x,0,s), add=TRUE, lty=1)
    legend('topright', legend=c('Empirical','Analytic'), lty=c(2,1))
    dev.off()
}


###############################################
# Problem 2
###############################################

### Sigma^2 = 1

#plot the risk in terms of b
pdf('hw1_2_plot_1.pdf')
curve(1000*x^2+(1-x)^2*1.08232,xlim=c(-.2,1.2),, ylim=c(0,2), xlab='b', ylab='Risk', mai
dev.off()

#simulate the risk
n<-1000
b<-100
s<-1
th.i<- function(i) rnorm(1,1/i^2,s)
JS<- function(n) max(1- n*s^2/sum(sapply(1:n,th.i)^2),0)
JSrisk<- function(n) {
    norm<-rnorm(n,0,s)
    theta<-1/((1:n)^2)
    z<-norm+theta
    js<-1- n*s^2/sum(z^2)
    return((js*z-theta)^2)
}
simrisk<-sapply(rep(n,times=b),JSrisk)
pdf('hw1_2_risk_1.pdf')
hist(simrisk, xlab='Simulated James-Stein Estimator Mean Squarred Error', ylab='Frequency
dev.off()
sim<-sapply(rep(n,times=b),JS)
mean<-format(mean(sim),trim=TRUE)
pdf('hw1_2_sim_1.pdf')
hist(sim, xlab='Simulated James-Stein Estimator', ylab='Frequency', main=paste('Performa
dev.off()


### sigma^2 = 1/n

#plot the risk in terms of b
pdf('hw1_2_plot_n.pdf')
```

```
curve(x^2+(1-x)^2*1.08232,xlim=c(-.2,1.2),, ylim=c(0,2), xlab='b', ylab='Risk', main='Ri
dev.off()

#simulate the risk
n<-1000
b<-100
s<-sqrt(1/n)
th.i<- function(i) rnorm(1,1/i^2,s)
JS<- function(n) 1- n*s^2/sum(sapply(1:n,th.i)^2)
JSrisk<- function(n) {
    norm<-rnorm(n,0,s)
    theta<-1/((1:n)^2)
    z<-norm+theta
    js<-1- n*s^2/sum(z^2)
    return((js*z-theta)^2)
}
simrisk<-sapply(rep(n,times=b),JSrisk)
pdf('hw1_2_risk_n.pdf')
hist(simrisk, xlab='Simulated James-Stein Estimator Mean Squarred Error', ylab='Frequency
dev.off()
sim<-sapply(rep(n,times=b),JS)
mean<-format(mean(sim),trim=TRUE)
pdf('hw1_2_sim_n.pdf')
hist(sim, xlab='Simulated James-Stein Estimator', ylab='Frequency', main=paste('Performa
dev.off()


#############################################
# Problem 3
#############################################

z<-read.csv(file="assn1-prob3-data.txt",head=FALSE)$V1

# This data set is known to be 1000 draws of N(0,1) plus value.
# My suspicion is that the means are draws from a pareto distribution with min ~2.5 and
# define function
rpareto<-function(n,a,m) m/(runif(n)^(1/a)) #random pareto vector
dpareto<-function(x,a,m) a*m^a/(x^(a+1))          #pareto density
den<-function(x,z,a,m) dpareto(x,a,m)*dnorm(z-x) #conditional density given z

# parameters
n<-1000
b<-25
al<-2
min<-2.4

# Simulate
p<-rnorm(1000)+rpareto(1000,al,min)
summary(p)
summary(z)

# QQ plot two simulated
pdf('hw1_3_qq.pdf')
```

```
qqplot(p,z, xlim=c(0,max(p,z)), main= 'Z versus Simulated', ylim=c(0,max(p,z)), xlab='Si
curve(x+0, add=TRUE, lty=1)
dev.off()

# Emperical versus simulated density
pdf('hw1_3_den.pdf')
plot(density(z), lty=2, ylab='Density', xlab='Z', main='Density of Z', xlim=c(-.5,75))
condden<-function(zp) integrate(function(p) den(p,zp,al,min), lower=min, upper=Inf, rel.
curve(sapply(x, condden), add=TRUE)
legend('topright', legend=c('Kernel Density Estimate of Z','Analytic Pareto + N(0,1)'),
dev.off()

# Maximize each estimate th
th<-rep(0, times=1000)
for (i in 1:1000) {
    scale=1/integrate(function(x) den(x,z[i],al,min),lower=min,upper=Inf, rel.tol=.Machi
    minfunc<-function(b) scale*integrate(function(x) (x-b)^2 * den(x,z[i],al,min),lower=
    th[i]<-optimize(f=minfunc,lower=0,c(z[i]/2,100))$minimum
}
th[z>30]<-(z-1/z)[z>30] # We run into numerical issues with large y, so we use stein

#Plot predicted theta versus Z
pdf('hw1_3_prediction.pdf')
plot(z,th, xlab='Original Z', ylab='Predicted Theta', main='Z vs. Predicted Thetas', xlin
curve(x+0, add=TRUE, lty=1)
dev.off()

# Output
write.table(th, file="assn1-prob3-ajmaurer.txt", row.names=FALSE, col.names=FALSE)


###############################################
# Problem 5
###############################################

doppler<-function(x) sqrt(x*(1-x))*sin(2.1*pi/(x+.05))
n<-1000
x<-(1:n)/n

# sigma = .1
y.1 <-sapply(x,doppler)+.1*rnorm(n)
gcv.1<-gcvplot(y.1~x,alpha=as.matrix((5:250)/1000))
pdf('hw1_5_p1_bandwith.pdf')
plot(gcv.1$alpha,gcv.1$values, main='CV Score, Sigma=.1', xlab='Bandwith', ylab='CV Scor
dev.off()

band.1 <- (gcv.1$alpha[order(gcv.1$values)])[1]
loc.1<-locfit(y.1~x,alpha=band.1)
crit(loc.1)<-crit(loc.1,cov=.95)
pdf('hw1_5_p1_fit.pdf')
plot(loc.1,band='local',main=paste('Doppler With Sigma=.1, Fit With Bandwith=',toString(
points(x,y.1, cex=.5)
dev.off()
```

```
# sigma = 1
y1 <-sapply(x,doppler)+rnorm(n)
gcv1<-gcvplot(y1~x,alpha=as.matrix((50:500)/1000))
pdf('hw1_5_1_bandwith.pdf')
plot(gcv1$alpha,gcv1$values, main='CV Score, Sigma=1', xlab='Bandwith', ylab='CV Score')
dev.off()

band1 <- (gcv1$alpha[order(gcv1$values)])[1]
loc1<-locfit(y1~x,alpha=band1)
crit(loc1)<-crit(loc1,cov=.95)
pdf('hw1_5_1_fit.pdf')
plot(loc1,band='local',main=paste('Doppler With Sigma=1, Fit With Bandwith=',toString(ba
points(x,y1, cex=.5)
dev.off()
```