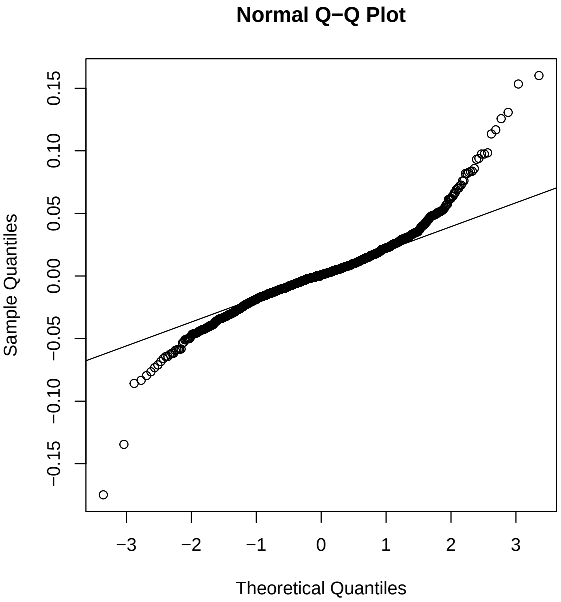
SOLUTIONS TO STATS 509 PROBLEM SET 3

1.

(a).

```
> Xday = read.csv('RecentFord.csv',header = TRUE)
> Fordday=Xday$Adj.Close
> Fordreturn = Fordday[-1]/Fordday[-length(Fordday)]-1
> # (a)
> qqnorm(Fordreturn)
> qqline(Fordreturn)
```



Since the right part of the QQ plot is above the qq line, the right tail of the data is heavier than that of normal. Since the left part of the QQ plot is above the qq line, the left tail of the data is heavier than that of normal. So in conclusion the distribution of the Ford returns is heavier tailed than normal.

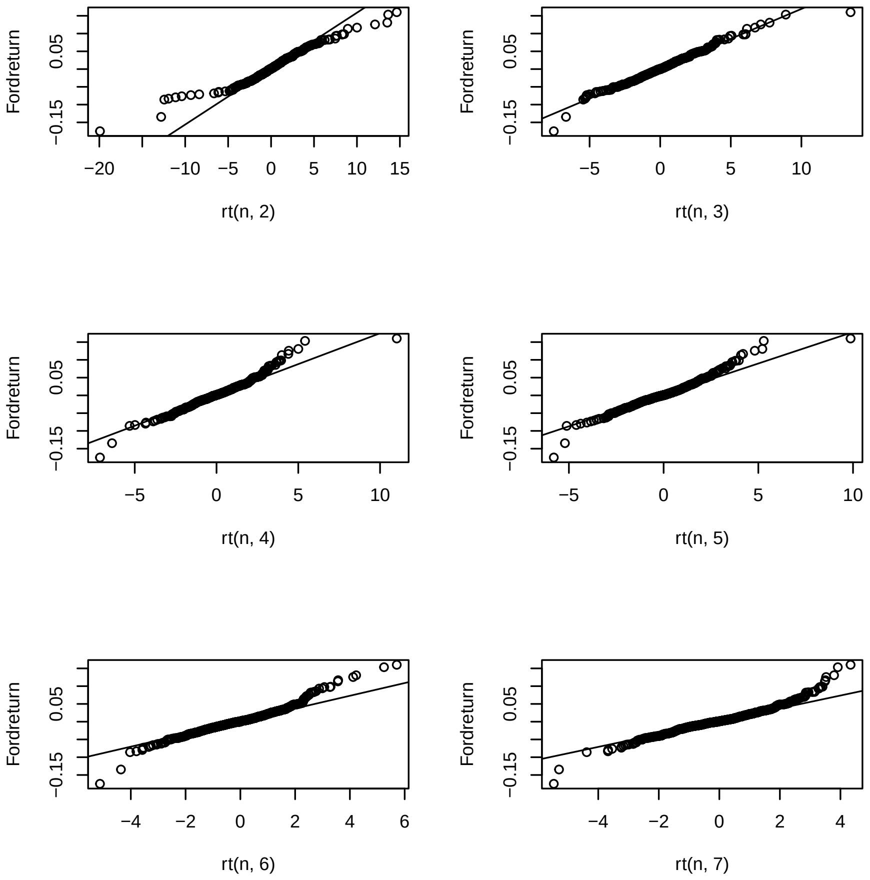
(b).

```
> # (b)
> shapiro.test(Fordreturn)

Shapiro-Wilk normality test

data:  Fordreturn
W = 0.93151, p-value < 2.2e-16
```

The p-value is essentially 0, and so we can statistically reject the normal distribution being appropriate for Ford returns. Of course with the sample size being 1257, it is not surprising that this is the case, and we would rely on the QQ plot for making a final judgement.

(c). Below are 6 QQ plots corresponding to 6 different degrees of freedom ranging from 2, 3, 4, 5, 6, and 7. It is clear that the distribution is lighter tailed than t-distn with 2 degrees of freedom, and it is heavier tailed than t-distn with 6 degrees of freedom. The t-distribution with 3 degrees of freedom appears to be the best match. It also looks like the -0.17 return data point doesn't affect the result too much. Dfferent realization may lead to different best degree of freedom and hence 3, 4, 5 are all accepted. And if based on your plot, if the -0.17 point is far away from the qq line, then you should consider it as an outlier and disregard it. As shown in the code, the -0:17 happens on May 12, 2009. The big news on Ford stock is here.

```
> # (c)
> n=length(Fordreturn)
> attach(mtcars)
> par(mfrow=c(3,2)) # setting up 2 x 2 arrangement of subplots
> qqplot(rt(n,2),Fordreturn)
> qqline(Fordreturn, distribution = function(x) {qt(x,df=2)})
> qqplot(rt(n,3),Fordreturn)
> qqline(Fordreturn, distribution = function(x) {qt(x,df=3)})
> qqplot(rt(n,4),Fordreturn)
> qqline(Fordreturn, distribution = function(x) {qt(x,df=4)})
> qqplot(rt(n,5),Fordreturn)
> qqline(Fordreturn, distribution = function(x) {qt(x,df=5)})
> qqplot(rt(n,6),Fordreturn)
> qqline(Fordreturn, distribution = function(x) {qt(x,df=6)})
> qqplot(rt(n,7),Fordreturn)
> qqline(Fordreturn, distribution = function(x) {qt(x,df=7)})
```



(d). In almost every Q-Q plot, the right tails is heavier and the left tail is lighter, which shows the asymmetries.

2.

(a). Let $w = (3.464) \cdot b$ be the width of the rectangular kernel. Now we have that the expected value of the kernel density estimate is given by

$$\begin{aligned} E\left(\hat{f}_b(x)\right) &= E\left\{\tfrac{1}{n}\textstyle\sum_{i=1}^{n} K_b\left(x - X_i\right)\right\} \\ &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n} E\left(K_b\left(x - X_i\right)\right) \\ &= E\left(K_b\left(x - X_1\right)\right) \\ &= \textstyle\int K_b\left(x - x'\right) f\left(x'\right) dx' \\ &= \tfrac{1}{w}\textstyle\int_{x-\frac{w}{2}}^{x+\frac{w}{2}} f\left(x'\right) dx' \\ &= \tfrac{1}{w}\left[F\left(x + \tfrac{w}{2}\right) - F\left(x - \tfrac{w}{2}\right)\right] \end{aligned}$$

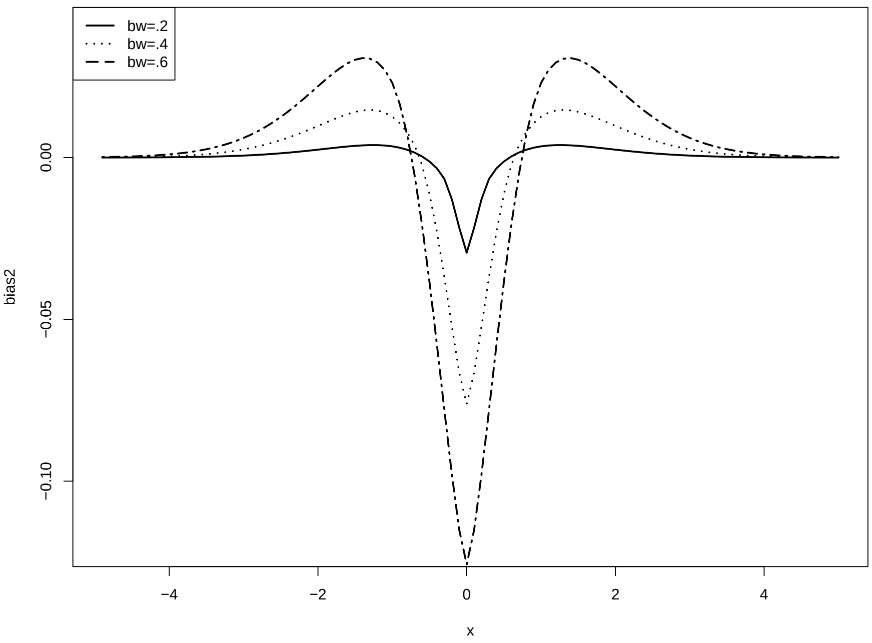
where F is the cdf of $\mathrm{GED}(0, 1, 1.5)$. The third line holds since $X_1, X_2, \ldots, X_n$ are identicially distributed, where f is pdf of $\mathrm{GED}(0, 1, 1.5)$

(b). The bias is given by

$$\text{bias}_b(x) = E\left(\hat{f}_b(x)\right) - f(x) = \frac{1}{w}\left[F\left(x + \frac{w}{2}\right) - F\left(x - \frac{w}{2}\right)\right] - f(x)$$

where f is the pdf of GED$(0, 1, 1.5)$. The plot of the bias function for bandwidths of .2, .4, and .6 are shown below along the R-code that generated the plots. We see as we expected that the overall level of bias (negative and positive) increases as the bandwidth increases. Also as expected, the bias is negative in the middle near to 0 (i.e., the kernel density tends to undershoot the main central peak), and then they all go positive outside of the main lobe at around the same value of approximately ±.8.

```
> # b
> library(fGarch)
> x = seq(-4.9,5,by=.1)
> b = .2;
> w = 3.464*b
> bias2 = (1/w)*(pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5)) - dged(x,0,1,1.5)
> b = .4;
> w = 3.464*b
> bias4 = (1/w)*(pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5)) - dged(x,0,1,1.5)
> b = .6;
> w = 3.464*b
> bias6 = (1/w)*(pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5)) - dged(x,0,1,1.5)
> dev.new()
NULL
> pdf('2b.pdf', width = 10, height = 8)
> plot(x,bias2,ylim = c(-.12,.04),type='l',lty=1,lwd=2)
> lines(x,bias4,lwd=2,lty=3)
> lines(x,bias6,lwd=2,lty=6)
> legend("topleft",c("bw=.2","bw=.4","bw=.6"),lwd=c(2,2,2),lty=c(1,3,5))
> dev.off()
RStudioGD
        2
```



(c). Similar to (a), for arbitrary sample size n, we have

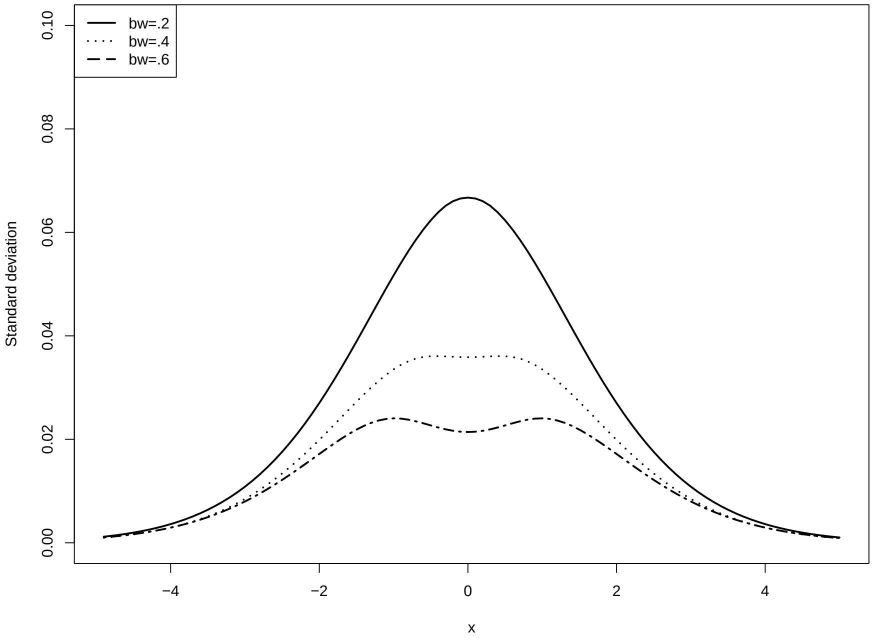
$$
\begin{aligned}
\mathrm{Var}[(\hat{f}_b(x))] &= \mathrm{Var}\left\{\frac{1}{n}\sum_{i=1}^{n} K_b(x-X_i)\right\} \\
&= \frac{1}{n}\mathrm{Var}(K_b(x-X_1)) \quad \text{by iid} \\
&= \frac{1}{n}\left[\mathbb{E}(K_b^2(x-X_1)) - \mathbb{E}^2(K_b(x-X_1))\right] \\
&= \frac{1}{n}\left[\int K_b^2(x-x')f(x')dx' - \mathbb{E}^2(\hat{f}_b(x))\right] \\
&= \frac{1}{n}\left[\frac{1}{\omega^2}\int_{-\frac{w}{2}}^{\frac{w}{2}} f(x)\,dx - \frac{1}{\omega^2}\left[F\left(x+\frac{w}{2}\right) - F\left(x-\frac{w}{2}\right)\right]^2\right] \\
&= \frac{1}{nw^2}\left[\left[F\left(x+\frac{w}{2}\right) - F\left(x-\frac{w}{2}\right)\right] - \left[F\left(x+\frac{w}{2}\right) - F\left(x-\frac{w}{2}\right)\right]^2\right]
\end{aligned}
$$

So

$$
\begin{aligned}
\mathrm{SD}(\hat{f}_b(x)) &= \sqrt{\mathrm{Var}[(\hat{f}_b(x))]} \\
&= \frac{1}{\sqrt{n}\omega}\sqrt{\left[F\left(x+\frac{w}{2}\right) - F\left(x-\frac{w}{2}\right)\right] - \left[F\left(x+\frac{w}{2}\right) - F\left(x-\frac{w}{2}\right)\right]^2}
\end{aligned}
$$

Now the final result will be to use this equation with $n = 100$, and the plots of the standard deviation for $b = .2, .4, .6$ are shown below.

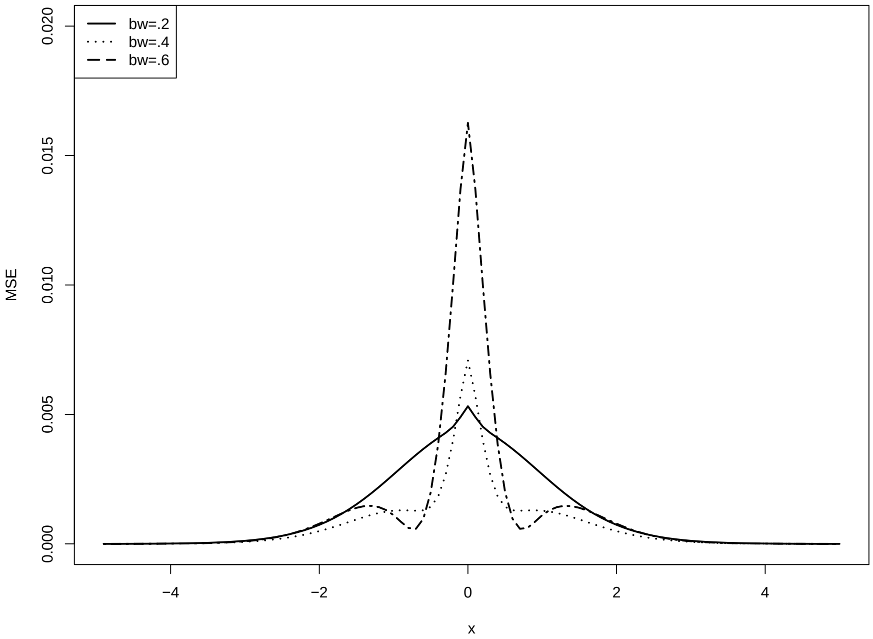
```
> # c
> library(fGarch)
> n = 100
> x = seq(-4.9,5,by=.1)
> b = .2;
> w = 3.464*b
> sd2 = (1/(w*sqrt(n)))*sqrt((pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5))  -
+                     ((pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5)) )^2)
> b = .4;
> w = 3.464*b
> sd4 = (1/(w*sqrt(n)))*sqrt((pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5))  -
+                               ((pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5)) )^2)
> b = .6;
> w = 3.464*b
> sd6 = (1/(w*sqrt(n)))*sqrt((pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5))  -
+                                     ((pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5)) )^2)
> dev.new()
NULL
> pdf('2c.pdf', width = 10, height = 8)
> plot(x,sd2,ylim = c(0,0.1),type='l',lty=1,lwd=2, ylab = 'Standard deviation')
> lines(x,sd4,lwd=2,lty=3)
> lines(x,sd6,lwd=2,lty=6)
> legend("topleft",c("bw=.2","bw=.4","bw=.6"),lwd=c(2,2,2),lty=c(1,3,5))
> dev.off()
RStudioGD
        2
```



(d). From (b) and (c), MSE is

$$\left\{\left(\frac{1}{w}\left[F\left(x+\frac{w}{2}\right)-F\left(x-\frac{w}{2}\right)\right]-f(x)\right)^2\right\}+$$
$$\frac{1}{n\omega^2}\left\{\left[F\left(x+\frac{w}{2}\right)-F\left(x-\frac{w}{2}\right)\right]-\left[F\left(x+\frac{w}{2}\right)-F\left(x-\frac{w}{2}\right)\right]^2\right\}$$

```
> # d
> library(fGarch)
> x = seq(-4.9,5,by=.1)
> b = .2;
> w = 3.464*b
> mse2 = bias2 ^ 2 + sd2 ^ 2
> b = .4;
> w = 3.464*b
> mse4 = bias4 ^ 2 + sd4 ^ 2
> b = .6;
> w = 3.464*b
> mse6 = bias6 ^ 2 + sd6 ^ 2
> dev.new()
NULL
> pdf(’2d.pdf’, width = 10, height = 8)
> plot(x,mse2,ylim = c(0,0.02),type=’l’,lty=1,lwd=2, ylab = ’MSE’)
> lines(x,mse4,lwd=2,lty=3)
> lines(x,mse6,lwd=2,lty=6)
> legend("topleft",c("bw=.2","bw=.4","bw=.6"),lwd=c(2,2,2),lty=c(1,3,5))
> dev.off()
RStudioGD
        2
```

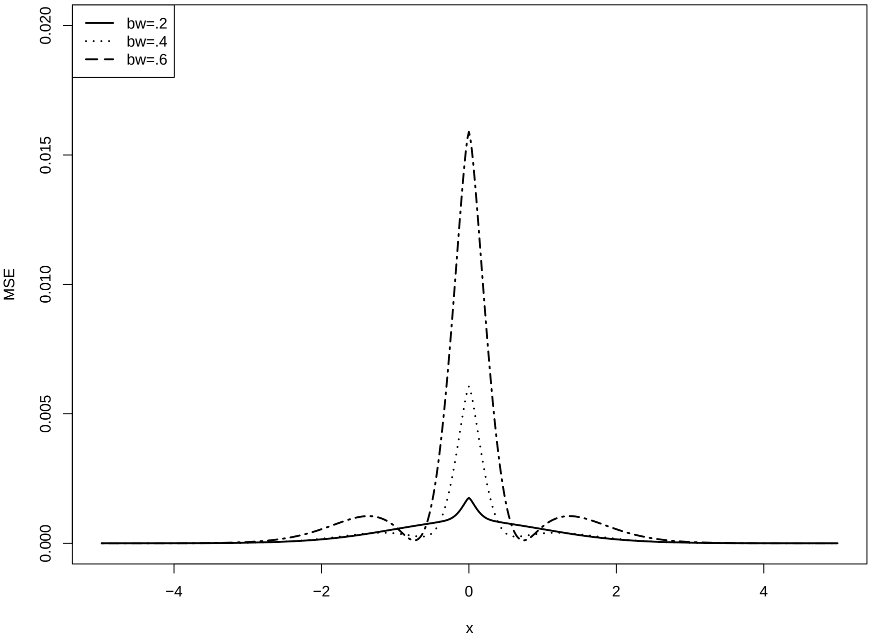


$bw = .4$ seems to be the best choice from the plot. Further integrating also confirms the case:

```
> # integrated MSE
> imse2 = 10 * sum(mse2) / length(x)
> imse2
[1] 0.01177077
> imse4 = 10 * sum(mse4) / length(x)
> imse4
[1] 0.007387609
> imse6 = 10 * sum(mse6) / length(x)
> imse6
[1] 0.01283861
```

(e).

```
> # e
> library(fGarch)
> n = 500
> x = seq(-4.98,5,by=.02)
> b = .2;
> w = 3.464*b
> mse2 = ((1/w)*(pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5)) - dged(x,0,1,1.5)) ^ 2 +
+        ((1/(w*sqrt(n)))*sqrt((pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5))  -
+                                ((pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5)) )^2)) ^ 2
> b = .4;
> w = 3.464*b
> mse4 = ((1/w)*(pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5)) - dged(x,0,1,1.5)) ^ 2 +
+   ((1/(w*sqrt(n)))*sqrt((pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5))  -
+                            ((pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5)) )^2)) ^ 2
> b = .6;
> w = 3.464*b
> mse6 = ((1/w)*(pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5)) - dged(x,0,1,1.5)) ^ 2 +
+   ((1/(w*sqrt(n)))*sqrt((pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5))  -
+                            ((pged(x+w/2,0,1,1.5)-pged(x-w/2,0,1,1.5)) )^2)) ^ 2
> dev.new()
NULL
> pdf('2e.pdf', width = 10, height = 8)
> plot(x,mse2,ylim = c(0,0.02),type='l',lty=1,lwd=2, ylab = 'MSE')
> lines(x,mse4,lwd=2,lty=3)
> lines(x,mse6,lwd=2,lty=6)
> legend("topleft",c("bw=.2","bw=.4","bw=.6"),lwd=c(2,2,2),lty=c(1,3,5))
> dev.off()
RStudioGD
        2
```



It is clear that now $bw = .2$ is a better choice.

```
> # integrated MSE
> imse2 = 10 * sum(mse2) / length(x)
> imse2
[1] 0.002556436
> imse4 = 10 * sum(mse4) / length(x)
> imse4
[1] 0.003782237
```

```
> imse6 = 10 * sum(mse6) / length(x)
> imse6
[1] 0.01094876
```