STAT 374 | HW1

Julian McClellan

April 23, 2016

Contents

1. Computing and plotting with R(15 points)

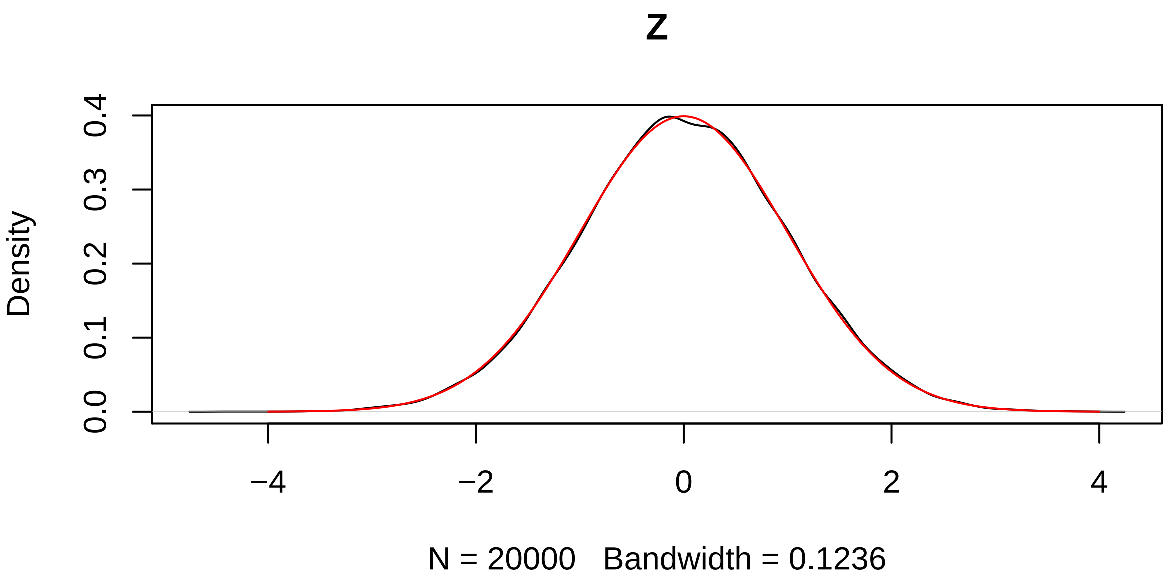
(a)

```
sample_size = c(seq(1, 200))
sim = function(part_b = FALSE, ssize = sample_size, sig = 1,
    b = 100) {
    rv = c()
    for (n in sample_size) {
        stat_n = c()

        for (rep in (1:b)) {
            mew_n = mean(rnorm(n, 1, sig))
            if (part_b) {
                stat_b = sqrt(n) * (mew_n - 1)  # Z
            } else {
                stat_b = (mew_n - 1)^2  # The MSE
            }
            stat_n = append(stat_n, stat_b)
        }

        if (part_b) {
            stat_rv = stat_n
        } else {
            stat_rv = mean(stat_n)
        }
        rv = append(rv, stat_rv)
    }
    if (part_b) {
        x = seq(-4, 4, 0.01)
        f = dnorm(x, sd = sig)
        plot(density(rv), main = "Z")
        lines(x, f, "l", col = "red")
    } else {
        # Normal scale plot
        plot(sample_size, rv, ylab = "")
        lines(sample_size, (sig^2)/sample_size, col = "red")
        title(expression(paste(frac(sigma^2, n), "vs. Empirical MSE")),
            line = -2)
```

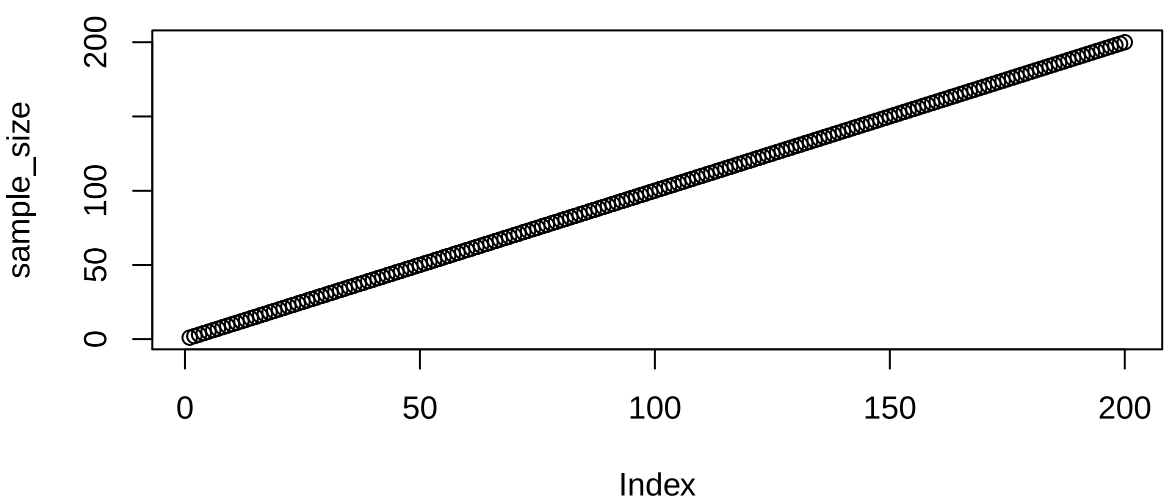
```
        # Log Log Plot
        plot(sample_size, log = "xy", rv, ylab = "log 1", xlab = "log sample size")
        lines(sample_size, (sig^2)/sample_size, col = "red")
        title(expression(paste("log(", frac(sigma^2, n), ") ",
            "vs. log(Empirical MSE)")), line = -2)
    }

}
```

(b)

```
zs = sim(TRUE)
```




```
plot(sample_size, zs)
```



2. Leave-one-out cross-validation (20 points)

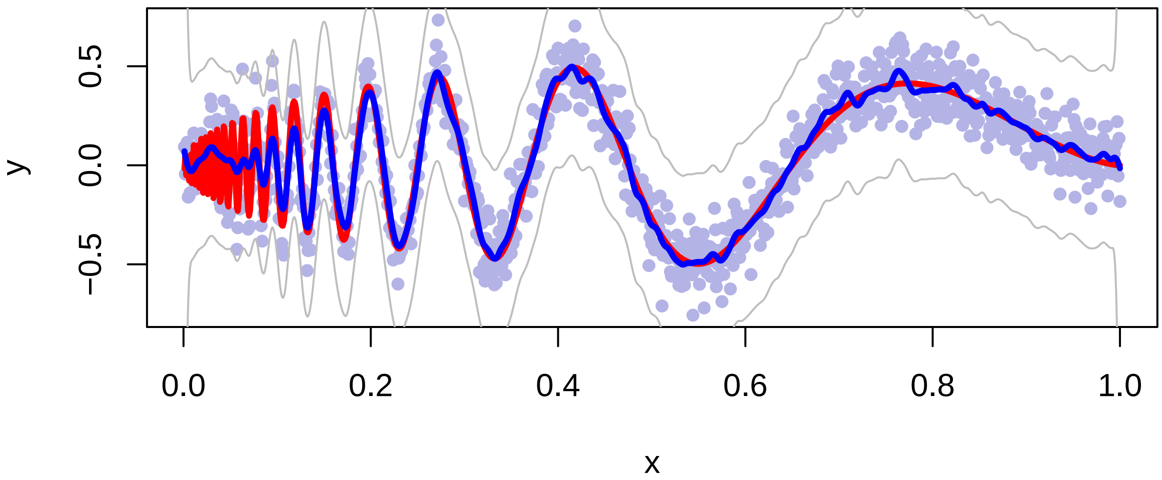
(a)

(b)

```r
doppler = function(x) {
    return(sqrt(x * (1 - x)) * sin(2.1 * pi/(x + 0.05)))
}

cvs = function(h, x, y) {
    llr = locfit(y ~ x, alpha = c(0, h), deg = 1)

    # From 'smoothdemo.r'
    Lii = predict(llr, where = "data", what = "infl")

    return(mean(((y - fitted(llr))/(1 - Lii))^2))
}

model = function(num_obs = 1000, sigma = 0.1) {
    # Generate data
    x = (1:num_obs)/num_obs
    y = doppler(x) + sigma * rnorm(num_obs)

    # Determine optimal h (using the function cvs as the 'formula derived in part
    # (a)')
    h = seq(0.01, 0.8, 0.01)
    cv_scores = unlist(lapply(h, cvs, x = x, y = y))
    hstar = h[cv_scores == min(cv_scores)]

    # Local linear regression fitted using locfit lib
    out = locfit(y ~ x, alpha = c(0, hstar), deg = 1)
    plot(x, y, pch = 16, cex = 0.9, col = rgb(0.7, 0.7, 0.9))
    lines(x, doppler(x), "l", col = "red", lwd = 3)
    lines(x, fitted(out), "l", col = "blue", lwd = 3)

    # From 'smoothdemo.r'
    normell = predict(out, where = "data", what = "vari")
    n = length(x)
    lines(x, fitted(out) + sqrt(n) * 2 * sigma * normell, "l", col = "gray", lwd = 1)
    lines(x, fitted(out) - sqrt(n) * 2 * sigma * normell, "l", col = "gray", lwd = 1)

}
# Run model
library(locfit)
```

```
## locfit 1.5-9.1     2013-03-22
```

```r
model()
```



Is $I_n(x)$ the 95 percent pointwise confidence interval for $r(x)$