STAT 579 Final

Yifan Zhu

November 12, 2016

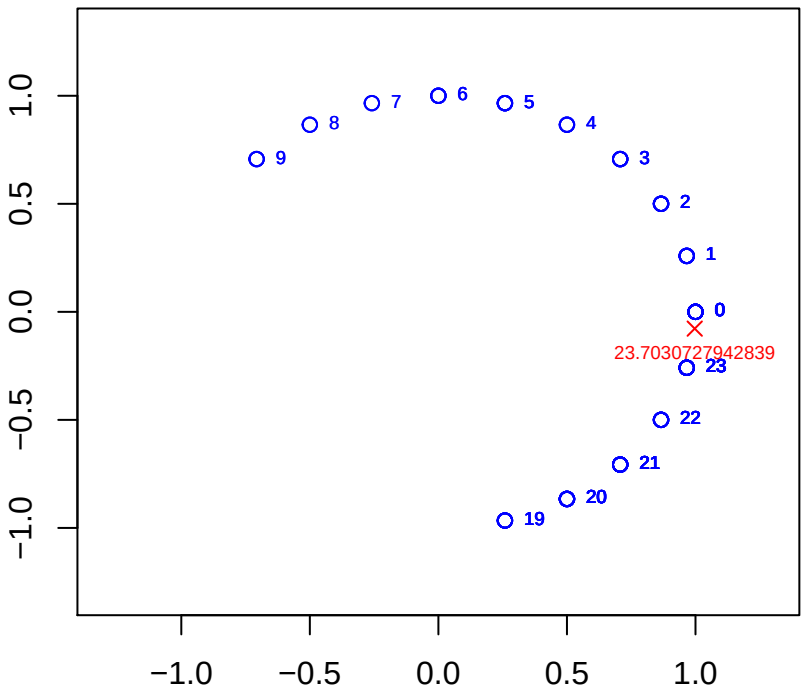
Problem 1

(a)

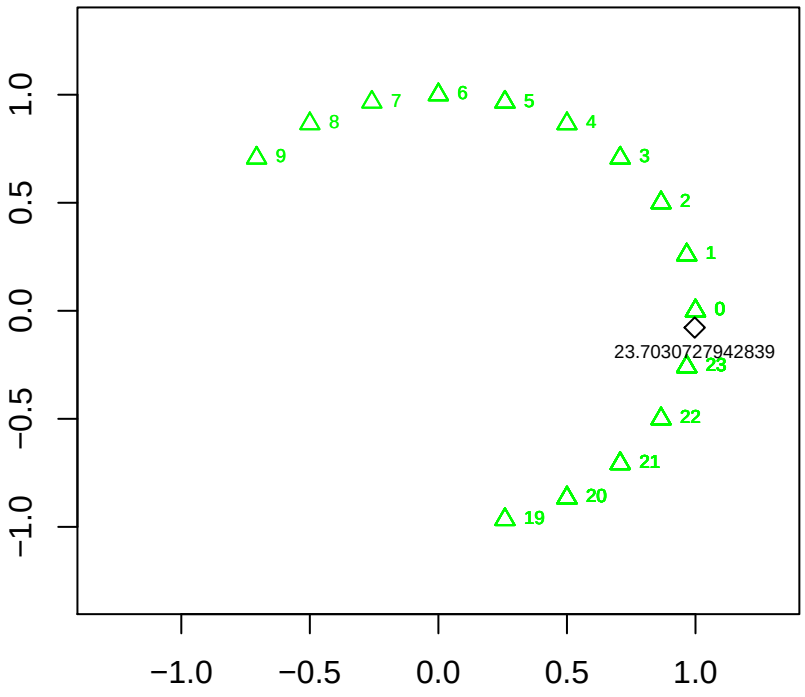
```
sleep <- read.table(file = "sleep.dat")

hour <- sleep$V1

circ.mean <- function(data, input.type) {

    circ.mean.rad <- function(data) {
        x <- cos(data)
        y <- sin(data)
        xbar <- mean(x)
        ybar <- mean(y)
        r <- sqrt(xbar^2 + ybar^2)
        cbar <- xbar/r
        sbar <- ybar/r

        thetabar <- atan2(y = sbar, x = cbar)
        ifelse(thetabar >= 0, thetabar, thetabar + 2 * pi)
    }


    if (input.type == "radian") {
        circ.mean.rad(data)
    } else {
        if (input.type == "degree") {
            data <- degree * pi/180
            degreebar.rad <- circ.mean.rad(data)
            degreebar <- degreebar.rad * 180/pi
            degreebar
        } else {
            if (input.type == "hour") {
                data <- data * pi/12
                hourbar.rad <- circ.mean.rad(data)
                hourbar <- hourbar.rad * 12/pi
                hourbar
            } else {
                cat("****** Wrong input.type ******")
            }
        }
    }
}
```

(b)

```
display.average <- function(data, col, pch, ...) {
    theta <- data * pi/12
    x <- cos(theta)
    y <- sin(theta)
    hourbar <- circ.mean(data = data, input.type = "hour")
    hourbartheta <- hourbar * pi/12
    xhourbar <- cos(hourbartheta)
    yhourbar <- sin(hourbartheta)

    plot(x = x, y = y, xlim = c(-1.3, 1.3), ylim = c(-1.3, 1.3), col = col[1],
        pch = pch[1], xlab = "", ylab = "", ...)
    text(x = x, y = y, data, col = col[1], cex = 0.6, pos = 4)
    points(x = xhourbar, y = yhourbar, col = col[2], pch = pch[2], ...)
    text(x = xhourbar, y = yhourbar, hourbar, col = col[2], pch = pch[2], cex = 0.6,
        pos = 1)
}

display.average(data = hour, col = c("blue", "red"), pch = c(1, 4))
```




```
display.average(data = hour, col = c("green", "black"), pch = c(2, 5))
```



(c) (i)

```
fhour <- function(hour, mu, kappa, log = FALSE) {
    theta <- hour * pi/12
    thetamu <- mu * pi/12
    f <- exp(kappa * cos(theta - thetamu))/(2 * pi * besselI(kappa, nu = 0))
    if (!log)
        f else log(f)
}
```

(ii)

```
# the log likelihood function l
l <- function(hour, mu, kappa) {
    sum(fhour(hour = hour, mu = mu, kappa = kappa, log = T))
}

# estimate the parameter mu (in hour)

mu <- circ.mean(hour, input.type = "hour")

mu
```

```
## [1] 23.70307
```

```
kappa <- optimize(f = l, mu = mu, hour = hour, maximum = T, interval = c(0.001,
    1000), lower = 0.001)

kappa
```

```
## $maximum
## [1] 1.652134
##
## $objective
## [1] -138.6694
```

(iii)

A.

```
rVM <- function(mu, kappa, n) {
    nuC <- cos(mu)
    nuS <- sin(mu)
    sigma <- 1/sqrt(kappa)
    X <- rnorm(n = n * 2, mean = 0, sd = 1)
    dim(X) <- c(n, 2)

    X <- X * sigma

    X[, 1] <- X[, 1] + nuC
    X[, 2] <- X[, 2] + nuS




    atan2(X[, 2], X[, 1])
}
```

B.

```
thetabar <- circ.mean(hour, input.type = "hour") * pi/12
teststat <- cos(thetabar - 0)
teststat
```

```
## [1] 0.9969801
```

C.

```
kappahat <- kappa$maximum

B <- 1000

simulationnum <- rep(0, B)

simulation <- lapply(simulationnum, FUN = rVM, kappa = kappahat, n = 100)
```

```r
teststat.sample <- lapply(simulation, FUN = function(x) cos(circ.mean(data = x,
    input.type = "radian") - 0))

pvalue <- sum(unlist(teststat.sample) > teststat)/B

pvalue
```

```
## [1] 0.61
```

The p-value is larger than 5%, thus there is evidence that the true mean direction $\mu = 0$.

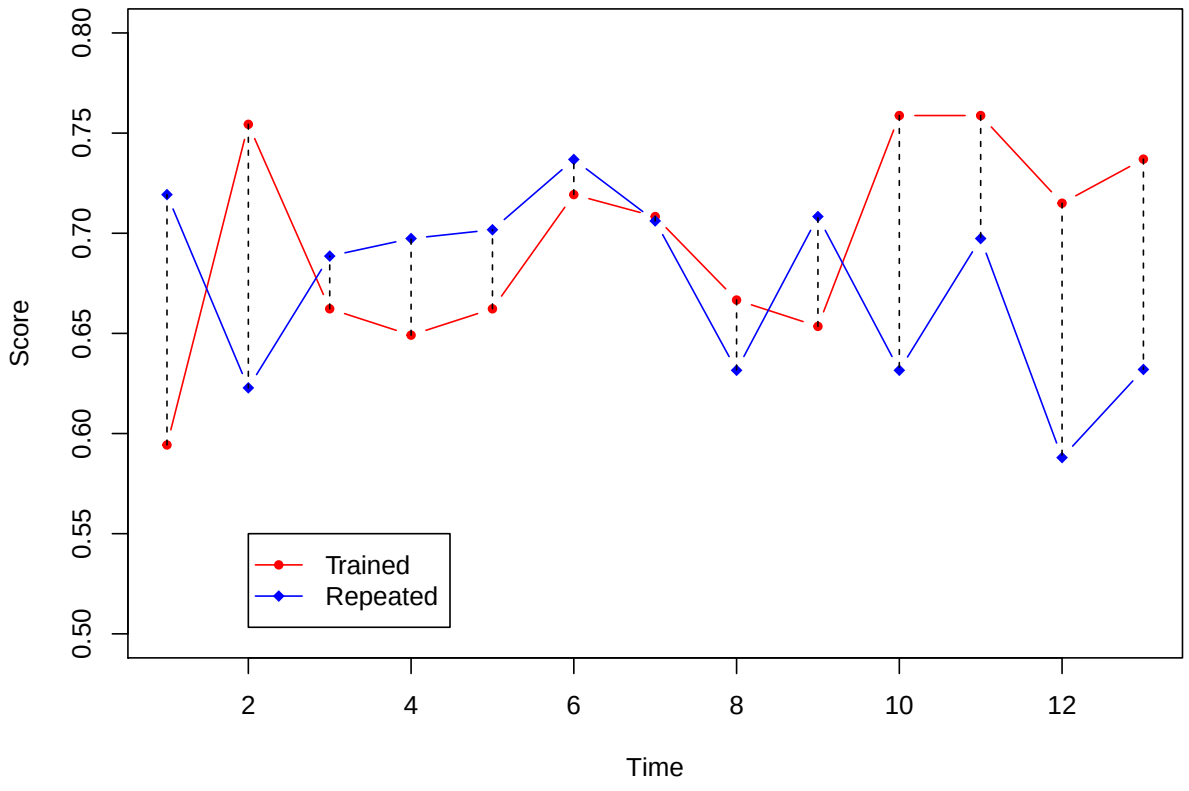
Problem 2

(a)

```
score <- read.csv(file = "brenda.csv")

plot(x = score$Time, y = score$Time, ylim = c(0.5, 0.8), "n", xlab = "Time",
    ylab = "Score")

lines(x = score$Time, y = score$Trained, col = "red", pch = 20, lty = 1, type = "b")

lines(x = score$Time, y = score$Repeated, col = "blue", pch = 18, lty = 1, type = "b")

apply(score, MARGIN = 1, FUN = function(x) lines(x = c(x[1], x[1]), y = c(x[2],
    x[3]), lty = 2))
```

```
## NULL
```

```
legend(2, 0.55, c("Trained", "Repeated"), col = c("red", "blue"), pch = c(20,
    18), lty = 1)
```



(b)

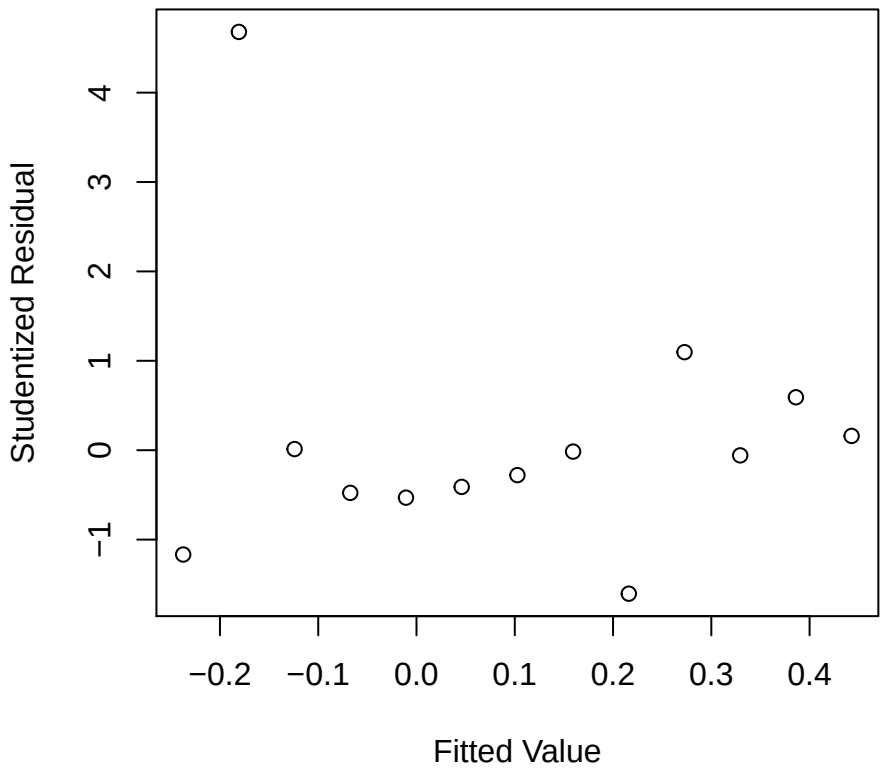
```
logit <- data.frame(Time = score$Time, Trained = log(score$Trained/(1 - score$Trained)),
    Repeated = log(score$Repeated/(1 - score$Repeated)))
```

(c)

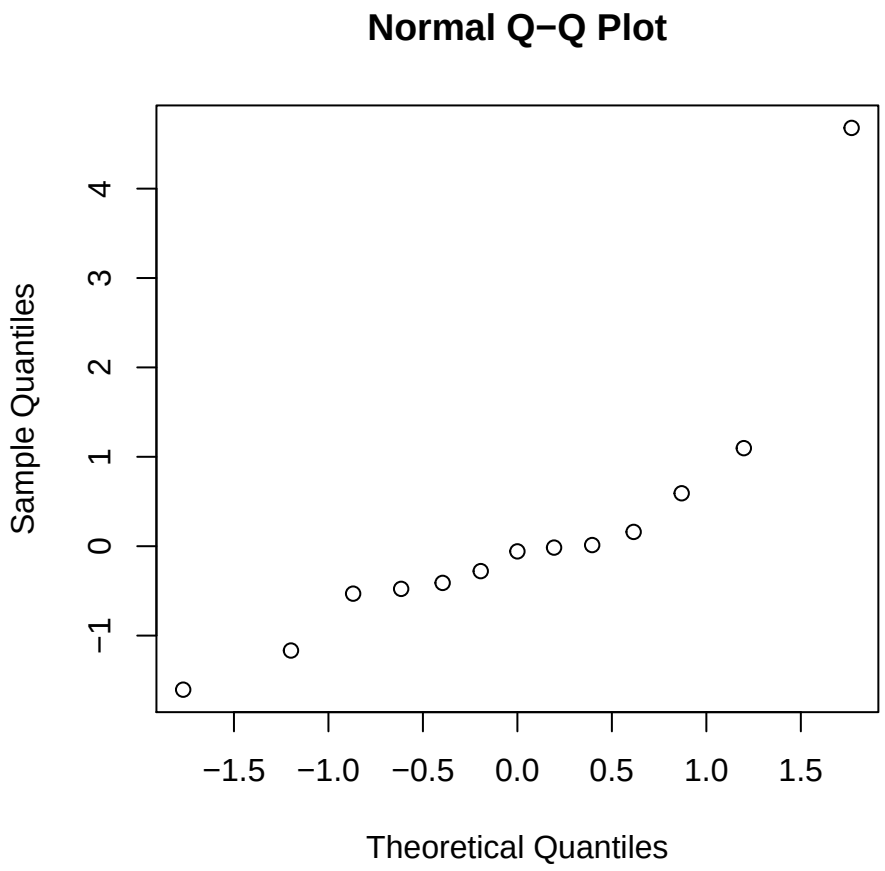
```
diff <- logit$Trained - logit$Repeated

logit <- data.frame(logit, Difference = diff)

lmest <- lm(formula = Difference ~ Time, data = logit)

summary(lmest)
```

```
##
## Call:
## lm(formula = Difference ~ Time, data = logit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46884 -0.15223 -0.01824  0.04685  0.80142
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.29412    0.19358  -1.519   0.1569
## Time         0.05668    0.02439   2.324   0.0403 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.329 on 11 degrees of freedom
## Multiple R-squared:  0.3293, Adjusted R-squared:  0.2684
## F-statistic: 5.402 on 1 and 11 DF,  p-value: 0.04028
```

```
plot(x = lmest$fitted.values, y = rstudent(lmest), xlab = "Fitted Value", ylab = "Studentized Residual"
```




```
require(graphics)
qqnorm(rstudent(lmest))
```



Time is significant in this model. The r-squared is around 30%, which is not very large.

The residuals seem to be smaller for the fitted values in $(-0.1, 0.2)$ than those for the fitted values in $(0.25, 0.45)$. The regression model is not so good for the data. And according to the Q-Q Plot the residuals are not normally distributed.