Hw2 S135

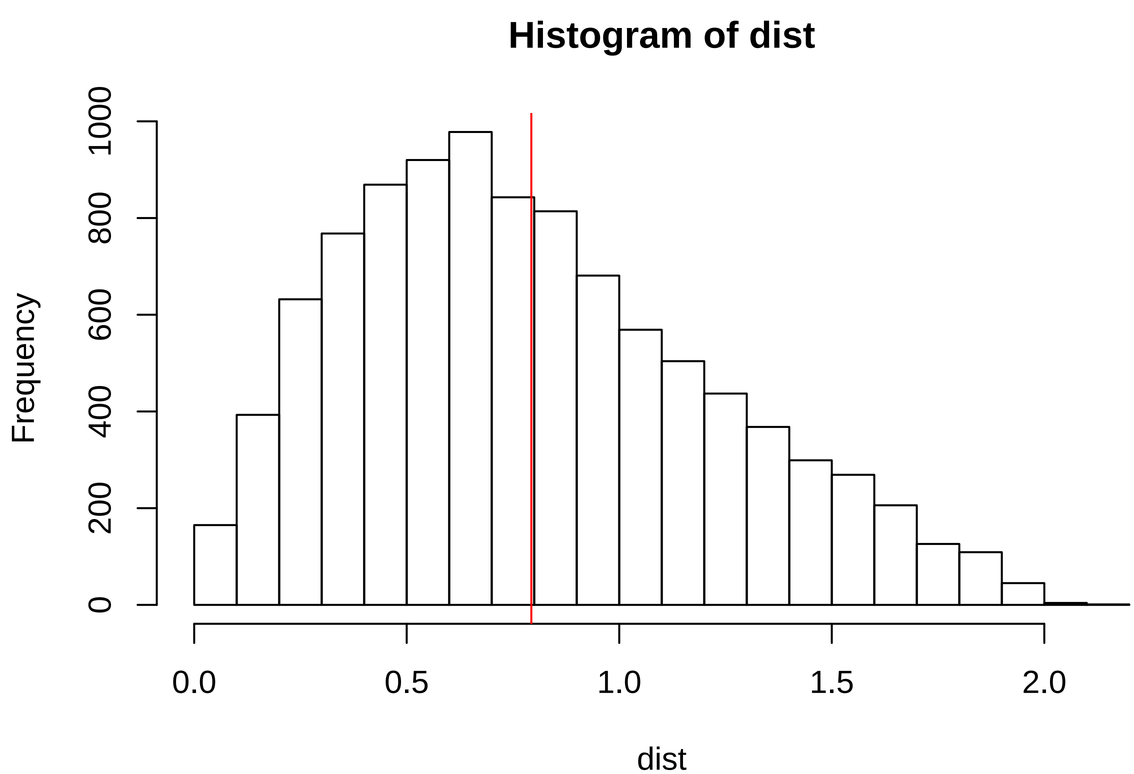
Callin Switzer

October 6, 2014

Homework #2 October 6, 2014

1. Select two points randomly in a rectangle of length=2 and width=1. Compute summary statistics & histogram as in class. In particular, what is the (estimated) mean distance between the points?

```
nsim <- 10000
set.seed(12345)
# get points
points <- data.frame(x1 = runif(n = nsim, min = 0, max = 2),
                     y1 = runif(n = nsim, min = 0, max = 1),
                     x2 = runif(n = nsim, min = 0, max = 2),
                     y2 = runif(n = nsim, min = 0, max = 1))
# calculate distances: sqrt((x2 - x1)^2 + (y2 - y1)^2)
dist <- with(points, sqrt((x2 - x1)^2 + (y2 - y1)^2))
hist(dist)
abline(v  = mean(dist), col = "red")
```

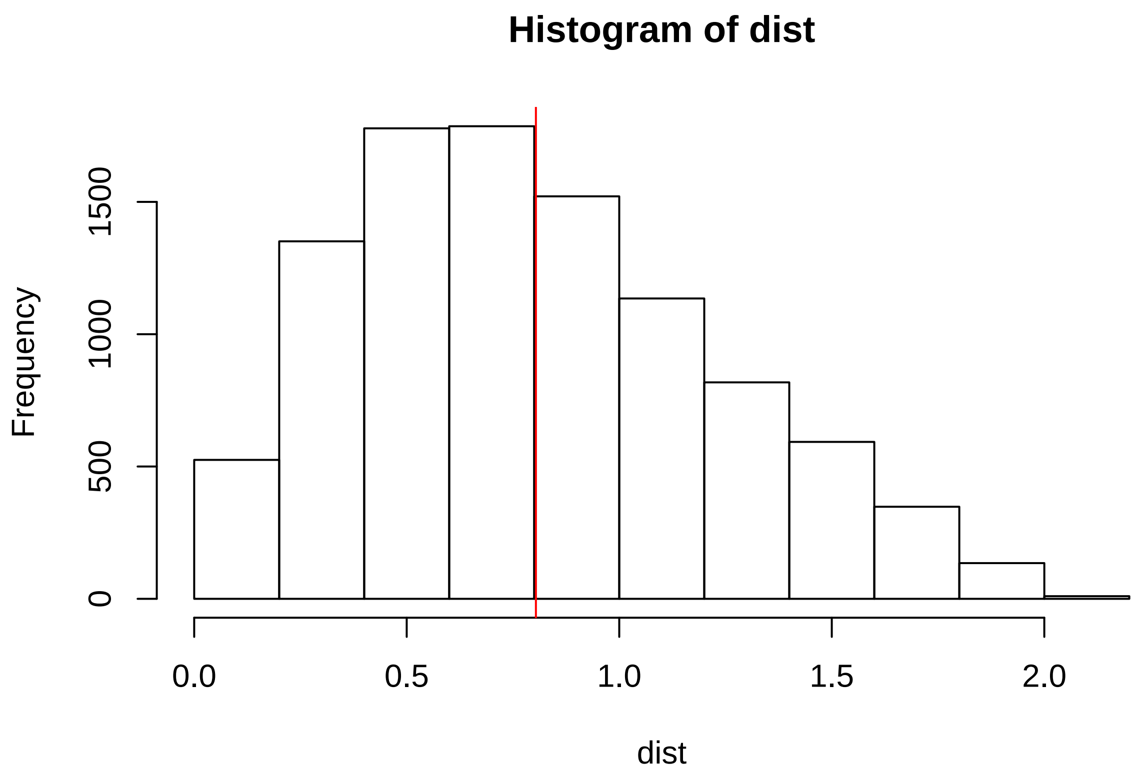



```
mean(dist)
```

```
## [1] 0.7932
```

2. Picture two unit squares sharing a side (like adjacent cells on a chessboard). Select a point randomly from each. What is the (estimated) mean distance between the points?

```
points <- data.frame(x1 = runif(n = nsim, min = 1, max = 2),
                     y1 = runif(n = nsim, min = 0, max = 1),
                     x2 = runif(n = nsim, min = 0, max = 2),
                     y2 = runif(n = nsim, min = 0, max = 1))

dist <- with(points, sqrt((x2 - x1)^2 + (y2 - y1)^2))
hist(dist)
abline(v  = mean(dist), col = "red")
```



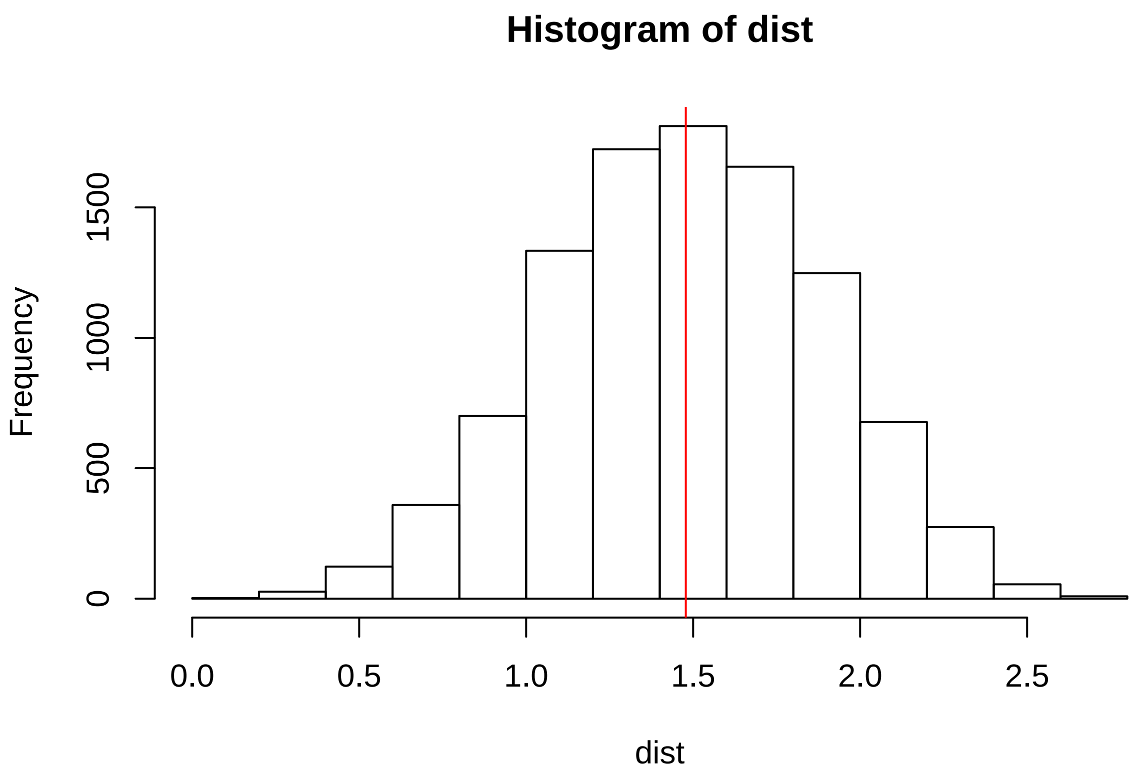

```
mean(dist)
```

```
## [1] 0.8039
```

3. Picture two unit squares sharing a corner and diagonal line (again, as on a chessboard). Select a point randomly from each.
   What is the (estimated) mean distance between the points?

```
nsim <- 10000

points <- data.frame(x1 = runif(n = nsim, min = 1, max = 2),
                     y1 = runif(n = nsim, min = 1, max = 2),
                     x2 = runif(n = nsim, min = 0, max = 1),
                     y2 = runif(n = nsim, min = 0, max = 1))

dist <- with(points, sqrt((x2 - x1)^2 + (y2 - y1)^2))
```

```r
hist(dist)
#plot(points$x2,points$y2, xlim = c(0,2), ylim = c(0,2))
#points(points$x1,points$y1, col = "red")
abline(v  = mean(dist), col = "red")
```




```r
mean(dist)
```

```
## [1] 1.478
```

4. Select two points randomly in the unit square. Draw a line segment connecting the two points. Now select two more points randomly in the unit square. Draw a new line segment connecting the two new points. Estimate the probability that the two line segments cross (that is, have a point in common).

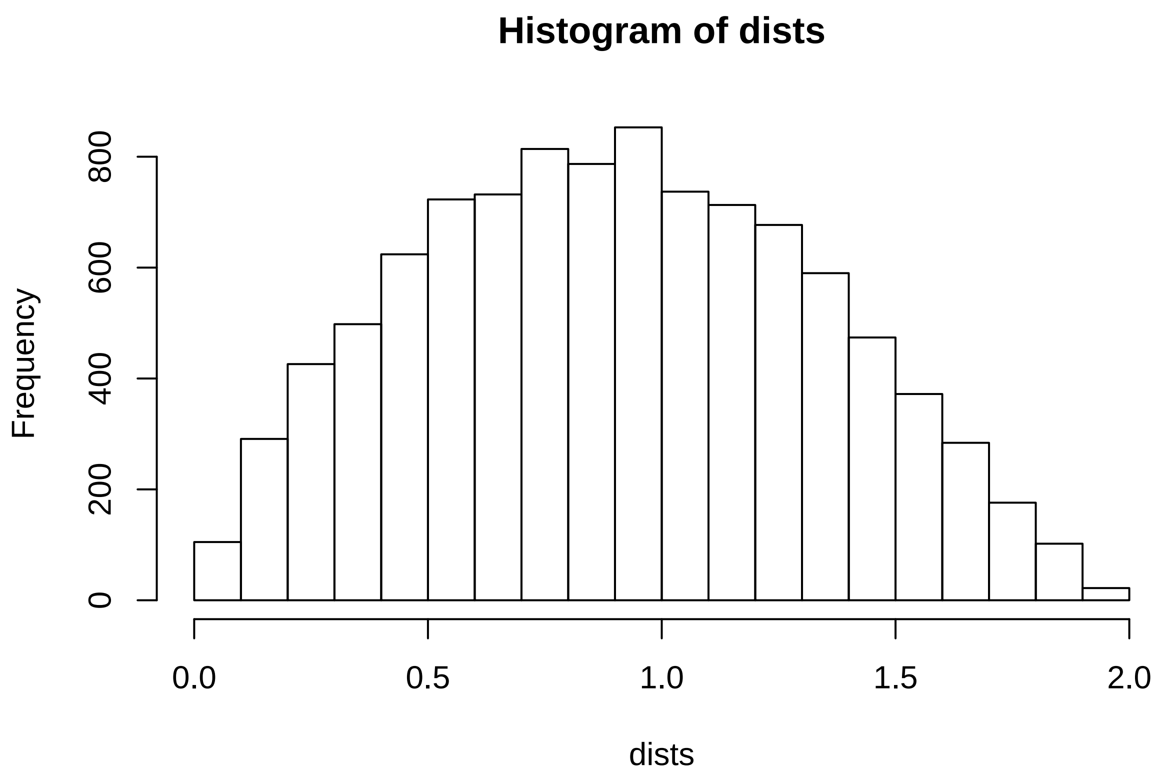
```r
# 2 points in unit square

linCross <- function(o) {
    # First two models
     df1 <- data.frame(x=runif(2), y=runif(2))
     m1 <- lm(y~x, df1) # slope
     df2 <- data.frame(x=runif(2), y=runif(2))
     m2 <- lm(y~x, df2) # slope

     # Plot them to show the intersection visually
     #plot(df1,xlim = c(0,1), ylim = c(0,1))
     #segments(df1$x[1], df1$y[1],df1$x[2], df1$y[2])
     #segments(df2$x[1], df2$y[1],df2$x[2], df2$y[2])
     #points(df2)
```

```
    # Now calculate it!
    a <- coef(m1)-coef(m2)
    cm <- rbind(coef(m1),coef(m2)) # Coefficient matrix
    foo <- c(-solve(cbind(cm[,2],-1)) %*% cm[,1])

    #points(foo[1], foo[2])

    ## Now, if we project the lines onto the x and y axis,
    ## the point of intersection must be between them

    #check xs
    xx1 <- max(df1$x[1], df1$x[2]) >= foo[1] & foo[1] >= min(df1$x[1], df1$x[2])
    #check xs on second line
    xx2 <- max(df2$x[1], df2$x[2]) >= foo[1] & foo[1] >= min(df2$x[1], df2$x[2])
    # check ys
    xx3 <- max(df1$y[1], df1$y[2]) >= foo[2] & foo[2] >= min(df1$y[1], df1$y[2])
    #check ys on second line
    xx4 <- max(df2$y[1], df2$y[2]) >= foo[2] & foo[2] >= min(df2$y[1], df2$y[2])

    sum(xx1, xx2, xx3, xx4) == 4
}

lc <- replicate(10000, linCross())
sum(lc) / 10000
```

```
## [1] 0.2344
```

5. Select two points randomly in the unit disk (radius=1). Compute summary statistics & histogram as in class. In particular, what is the (estimated) mean distance between the points?

```
ddist <- function(o){
    nsim <- 2
    rdisk <- function(q)        # how to generate random points in the unit disk?
    {
      R <- matrix(runif(nsim), ncol=1)
      Theta <- matrix(runif(nsim,0,2*pi), ncol=1)
      XY <- cbind(R^q*cos(Theta),R^q*sin(Theta))
      #x11(width=5,height=5.5)    # keep old plot and open a new plotting window
      #plot(XY,xlim=c(-1,1),ylim=c(-1,1))
      #upp.crcl <- function(x) sqrt(1-x^2)
      #low.crcl <- function(x) -sqrt(1-x^2)
      #curve(upp.crcl,from=-1,to=1,add=T)
      #curve(low.crcl,from=-1,to=1,add=T)
      return(XY)
    }
    XY <- rdisk(1/2)
    return( with(points, sqrt((XY[2,1] - XY[1,1])^2 + (XY[2,2] - XY[1,2])^2)) )
}

dists <- replicate(10000, ddist())
hist(dists)
```




```
mean(dists)
```

```
## [1] 0.9085
```

6. Select three points randomly in the unit disk (radius=1). Compute summary statistics & histogram as in class. In particular, estimate the probability that the triangle is obtuse.

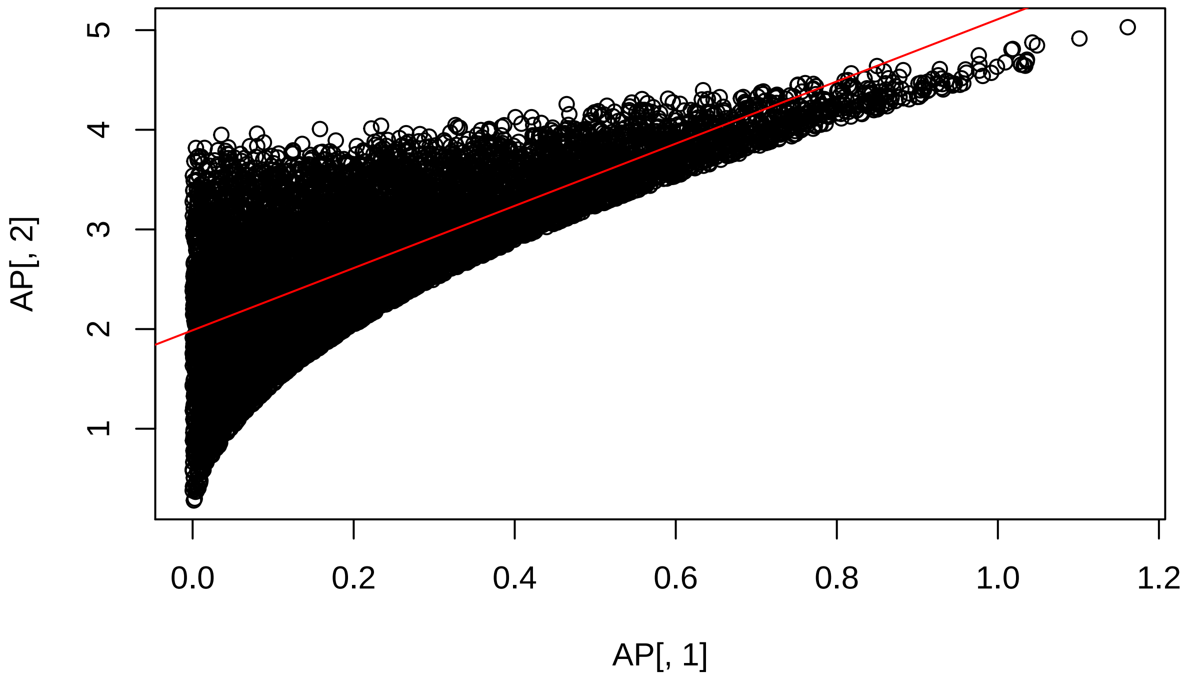
```
nsim <- 3
rdisk <- function(q)          # how to generate random points in the unit disk?
{
  R <- matrix(runif(nsim), ncol=1)
  Theta <- matrix(runif(nsim,0,2*pi), ncol=1)
  XY <- cbind(R^q*cos(Theta),R^q*sin(Theta))
  #x11(width=5,height=5.5)     # keep old plot and open a new plotting window
  #plot(XY,xlim=c(-1,1),ylim=c(-1,1))
  #upp.crcl <- function(x) sqrt(1-x^2)
  #low.crcl <- function(x) -sqrt(1-x^2)
  #curve(upp.crcl,from=-1,to=1,add=T)
  #curve(low.crcl,from=-1,to=1,add=T)
  return(XY)
}
angl <- function(v,w) acos(v%*%w/sqrt((v%*%v)*(w%*%w)))
                 # dot (inner) product denoted by %*%

#XY <- rdisk(1/2) # here are the three points

#plot(XY, type = "l")
```

```r
mxangl <- function(XY)
{
             #XY <- matrix(runif(6), ncol=2)
             A1 <- angl(XY[2,]-XY[1,],XY[3,]-XY[1,])
             A2 <- angl(XY[3,]-XY[2,],XY[1,]-XY[2,])
             A3 <- angl(XY[1,]-XY[3,],XY[2,]-XY[3,])
             max(A1,A2,A3) * (180/pi)

}

obt <- replicate(10000, mxangl(rdisk(1/2)) > 90)
sum(obt)/length(obt)
```

```
## [1] 0.7157
```

7. What is the cross-correlation between area & perimeter of random triangles in the unit disk? Perform a linear regression as in class.

```r
# generate three points and calculate area vs. perimeter
side <- function(u) sqrt(u%*%u)
angl <- function(v,w) acos(v%*%w/sqrt((v%*%v)*(w%*%w)))

rdiskAP <- function(q){
     nsim <- 3
     R <- matrix(runif(nsim), ncol=1)
     Theta <- matrix(runif(nsim,0,2*pi), ncol=1)
     XY <- cbind(R^q*cos(Theta),R^q*sin(Theta))
     S1 <- side(XY[3,]-XY[2,])
     S2 <- side(XY[1,]-XY[3,])
     S3 <- side(XY[2,]-XY[1,])
     A1 <- angl(XY[2,]-XY[1,],XY[3,]-XY[1,])
     area <- (1/2)*S2*S3*sin(A1)
     peri <- S1+S2+S3
     c(area=area,peri=peri)
}

#calculate area
AP <- t(replicate(10000, rdiskAP(1/2)))

plot(AP[,1], AP[,2])
modAP <- lm(AP[,2]~AP[,1])
abline(modAP, col = "red")
```




```
sm <- summary(modAP)
sqrt(sm$r.squared) # here is the correlation
```

```
## [1] 0.7603
```

8. Select four points randomly in a disk. With probability 1, no three of the points are collinear, so the convex hull of the four points is either a triangle (one point “inside” the others) or a quadrilateral. Estimate the probability that the convex hull is a triangle.

```
cnvx <- function(n)           # convex hull of n points in the unit disk (radius=1)
{
  R <- matrix(runif(n), ncol=1)
  Theta <- matrix(runif(n,0,2*pi), ncol=1)
  XY <- cbind(sqrt(R)*cos(Theta),sqrt(R)*sin(Theta))
  #x11(width=5,height=5.5)     # keep old plot and open a new plotting window
  #plot(XY,xlim=c(-1,1),ylim=c(-1,1),pch=22)
  h <- chull(XY)
  h <- c(h, h[1])
  #lines(XY[h, ],col='red',lwd=3)
  #upp.crcl  <- function(x) sqrt(1-x^2)
  #low.crcl  <- function(x) -sqrt(1-x^2)
  #curve(upp.crcl,from=-1,to=1,add=T)
  #curve(low.crcl,from=-1,to=1,add=T)
  length(unique(h)) == 4
}

chp <- replicate(10000, cnvx(4))
1 - sum(chp)/length(chp) # probability that it's a triangle
```

```
## [1] 0.2996
```

9. Do the same as in the previous problem, except select the four points randomly in an isosceles right triangular region.
10. Ask an original question about geometric probability, and answer it (approximately) using R's simulation capabilities.