GR5065 Assignment 2

Juan Lopez Martin

1 Darts

```r
score <- function(radius, angle) {
  stopifnot(is.numeric(radius), length(radius) == 1, radius >= 0)
  stopifnot(is.numeric(angle), length(angle) == 1,
            angle >= -2 * pi, angle < 2 * pi)

  if (radius > 170) return(0) # misses dartboard
  if (radius <= 6.35) return(50) # double bullseye
  if (radius <= 15.9) return(25) # single bullseye

  margin <- pi / 20
  interval <- margin * 2

  S <- darts:::getConstants()$S # 20, 1, ..., 5
  small <- pi / 2 - margin - 0:19 * interval
  large <- pi / 2 + margin - 0:19 * interval
  bed <- which(angle > small & angle <= large)
  if (length(bed) == 0) {
    angle <- angle - 2 * pi
    bed <- which(angle > small & angle <= large)
  }
  score <- S[bed]
  if (between(radius, 99, 107)) score <- 3 * score # in triple ring
  else if (between(radius, 162, 170)) score <- 2 * score # in double ring
  return(score)
}
```

```r
car_to_polar <- function(x, y){
  r <- sqrt(x^2 + y^2)
  theta <- atan2(y, x) # to get on 360 we have to *180/pi
  #theta <- ifelse(theta>0, theta, 360+theta)
  return(c(r, theta))
}

polar_to_car <- function(r, theta){
  x <- r*cos(theta)
  y <- r*sin(theta)
  return(c(x, y))
}
```

1.1 Prior Predictive Distribution

As a prior, let's say the median of sigma is at 50, with lower quartile at 35 and upper quartile at 65.

```r
library(rstan)
```

```
## Loading required package: StanHeaders

## Loading required package: ggplot2

## rstan (Version 2.19.2, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)

## For improved execution time, we recommend calling
## Sys.setenv(LOCAL_CPPFLAGS = '-march=native')
## although this causes Stan to throw an error on a few processors.
```

```r
expose_stan_functions("quantile_functions.stan")
args(JQPDS_icdf)
```

```
## function (p, lower_bound, alpha, quantiles, pstream__ = <pointer: 0x000000006b0c1540>)
## NULL
```

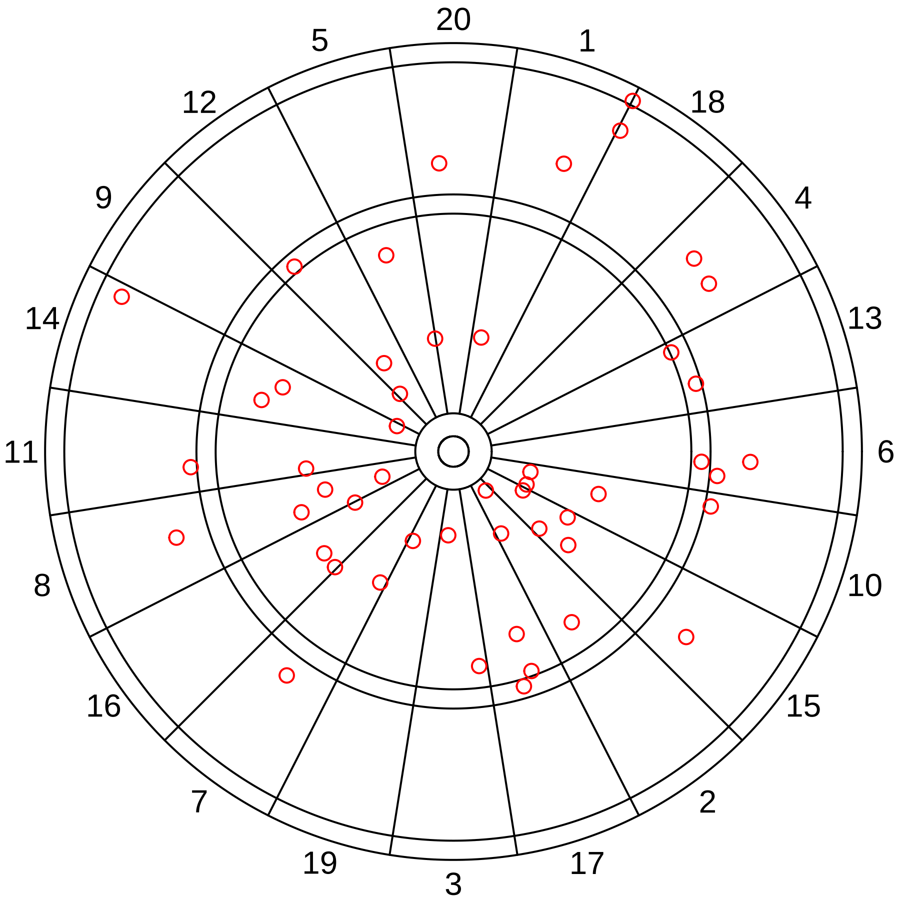
```r
prior_info <- c("1st Qu." = 35, "Median" = 50, "3rd Qu." = 65)
ps <- seq(0, 1, 0.01)

sigma_tilde <- sapply(ps, JQPDS_icdf, lower_bound = 0, alpha = 0.25, quantiles = prior_info)

#plot(ps, sigma_tilde, type = "l", ylab = "Prior CDF for sigma", xlab = "p")
```

```r
p = runif(1)
prior_info <- c("1st Qu." = 35, "Median" = 50, "3rd Qu." = 65)
sigma_tilde <- JQPDS_icdf(p = p, lower_bound = 0, alpha = 0.25, quantiles = prior_info)

print(paste("This implies the probablity of obtaining a sigma of ", round(sigma_tilde, 2), " or lower u
```

```
## [1] "This implies the probablity of obtaining a sigma of  61.06  or lower under the prior distributi
```

We assume $x \sim N(0, \tilde{\sigma})$ and $y \sim N(0, \tilde{\sigma})$.

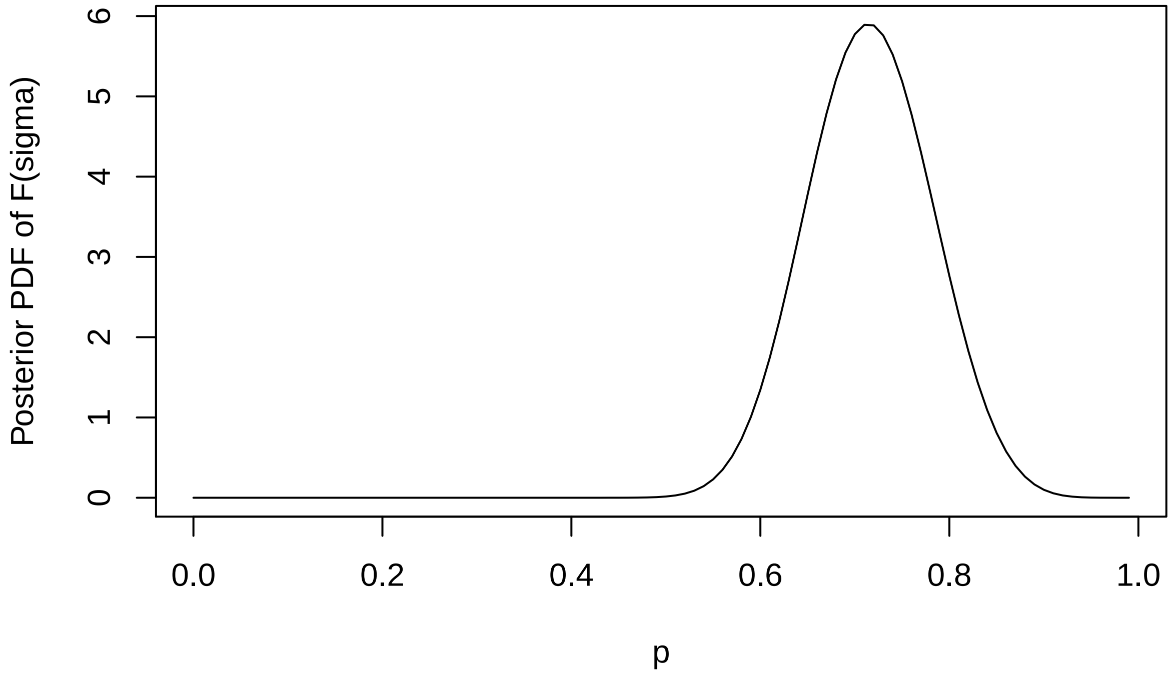
```r
x <- rnorm(50, 0, sigma_tilde)
y <- rnorm(50, 0, sigma_tilde)

drawBoard(new = TRUE)
points(x = x, y = y, col = 'red')
```



1.2 Posterior Distribution

Given $\mathcal{L}(\sigma; x, y) = p(x, y|\sigma)$. As we know x and y are independent, $p(x, y|\sigma) = p(x|\sigma)p(y|\sigma)$. Knowing the distributions of x and y we can express this (in R notation, to avoid excessive formulas) as `dnorm(x, 0, sigma) * dnorm(y, 0, sigma)`. As we have many data points, we must calculate $\mathcal{L}(\sigma; \mathbf{x}, \mathbf{y}) = \prod_{i=1}^{N} \mathcal{L}(\sigma; x_i, y_i)$.

```
numerator <- function(p_vec, x, y, quantiles) {
  return(sapply(p_vec, FUN = function(p) {
    sigma_tilde <- JQPDS_icdf(p = p, lower_bound = 0, alpha = 0.25, quantiles = quantiles)
    lik = dnorm(x, 0, sigma_tilde)*dnorm(y, 0, sigma_tilde)
    return(prod(lik))
  }))
}

p_vec <- seq(0, 0.99, length.out = 100)

denominator <- integrate(numerator, x = x, y = y, lower = 0, upper = 1, quantiles = prior_info)$value

plot(p_vec, numerator(p_vec, x, y, prior_info)/denominator, type = "l", xlab="p", ylab = "Posterior PDF
```

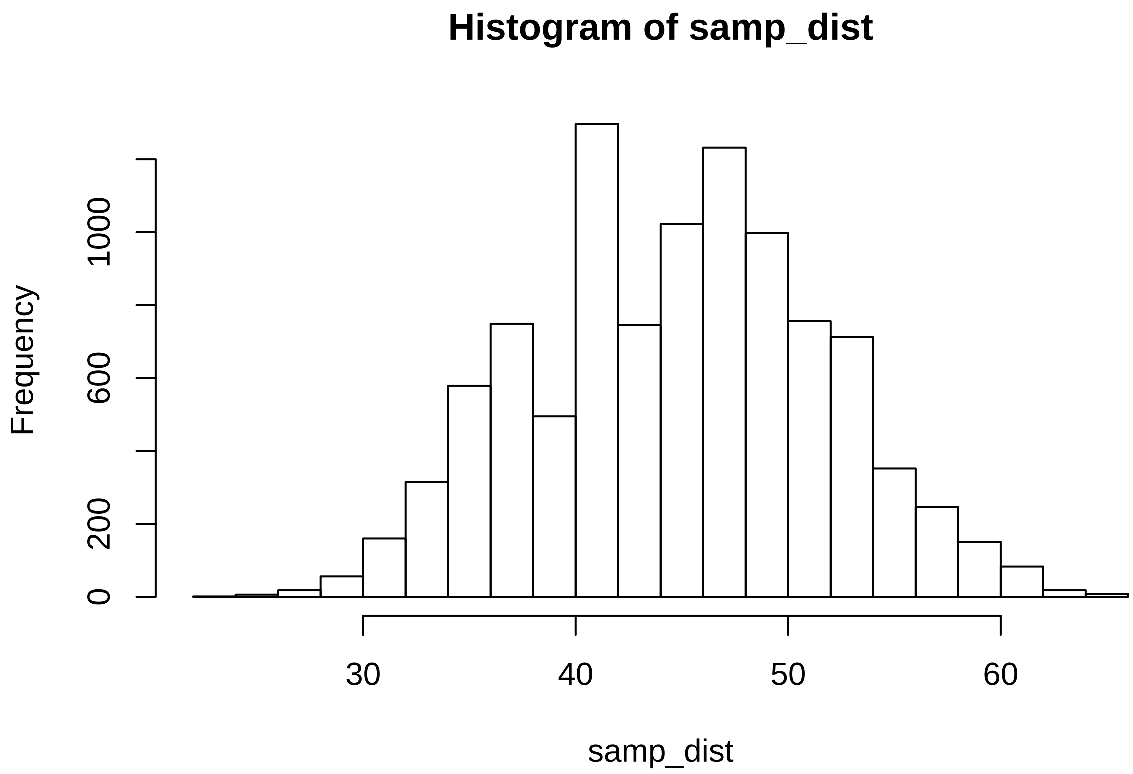


1.3 Sampling distribution for $\hat{\sigma}$

Tibshirani's estimator seems to be accurate, particularly considering we are only observing 50 scores.

```
sigma <- 46

sampling_distribution <- function(S, N, sigma) {
  sigma_hat <- rep(NA, S)
  for (s in 1:S) {
    # draw N values of x and y and
    x <- rnorm(N, 0, sigma)
    y <- rnorm(N, 0, sigma)

    # convert each of them to a radius and an angle
    r <- matrix(car_to_polar(x, y), ncol = 2)[,1]
    theta <- matrix(car_to_polar(x, y), ncol = 2)[,2]

    # call the score() function on each radius and angle
    scores <- mapply(score, r, theta)

    # call the simpleEM() function on the N scores
    sigma_squared <- simpleEM(scores)$s.final

    # put the square root of the $s.final into sigma_hat[s]
    sigma_hat[s] <- sqrt(sigma_squared)
  }
  return(sort(sigma_hat))
}
```

```
samp_dist <- sampling_distribution(S = 10000, N = 50, sigma = 46)
hist(samp_dist)
```



1.4 Hypothesis test

```
x = c(12,16,19,3,17,1,25,19,17,50,18,1,3,17,2,2,13,18,16,2,25,5,5,
1,5,4,17,25,25,50,3,7,17,17,3,3,3,7,11,10,25,1,19,15,4,1,5,12,17,16,
50,20,20,20,25,50,2,17,3,20,20,20,5,1,18,15,2,3,25,12,9,3,3,19,16,20,
5,5,1,4,15,16,5,20,16,2,25,6,12,25,11,25,7,2,5,19,17,17,2,12)
```

Note that the likelihood ratio is small if the alternative model is better than the null model. Thus: * Accept: If $\lambda > c$, the null model is a better fit to the data than the alternative. * Reject: If $\lambda < c$, the alternative model is a better fit to the data than the null.

In this case, we consider the null model as the fact that it was Tibshirani throwing the darts ($\sigma = 65$), while the alternative model corresponds to Price ($\sigma = 27$).

```
h0 <- 65 # Tibshirani
h1 <- 27 # Price

# Calculate lambda
loglik_tib <- simpleEM(x, s.init = h0^2)$loglik[1]
loglik_pri <- simpleEM(x, s.init = h1^2)$loglik[1]
lambda <- exp(loglik_tib - loglik_pri) # Note that lik1/lik2 = exp(loglik1) / exp(loglik2) = exp(loglik
# Calculate c from the previous exercise
sampl <- sampling_distribution(S = 1000, N = length(x), sigma = h0)
c <- quantile(sampl, probs = c(0.05))

print(paste(lambda, c, "Reject Null:", lambda < c))
```

```
## [1] "2.37903565314301e-06 56.1148611951493 Reject Null: TRUE"
```

We reject the null, meaning the alternative hypothesis appears to be more consistent witth the data. That is, considering those scores it is more likely that Price was the one throwing the darts.

Another way of seeing this problem is using `simpleEM` to estimate the σ of the player that got the scores contained in `x`.

```
sqrt(simpleEM(x)$s.final)
```

```
## [1] 28.26708
```

This $\sigma \approx 28$ is more similar to $\sigma_{Price} = 27$ that to $\sigma_{Tibshirani} = 65$.

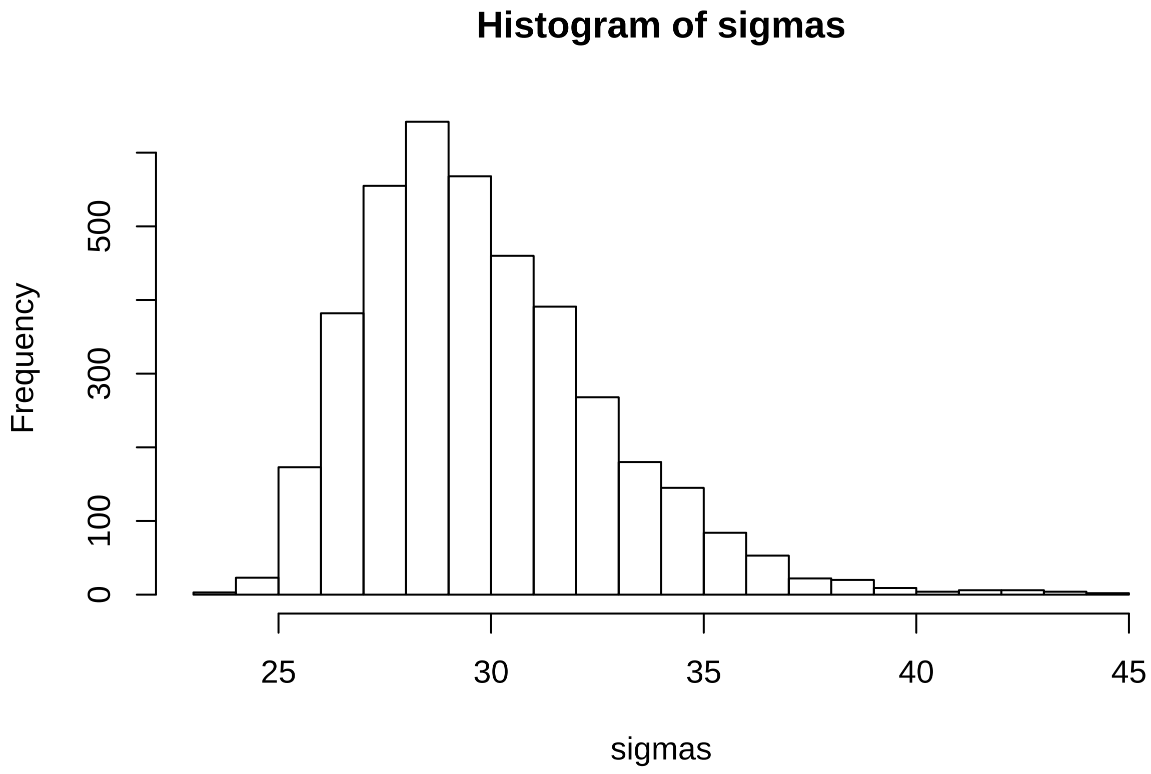
1.5 Posterior Distribution

```
# Warning: it takes a long time to run

x = c(12,16,19,3,17,1,25,19,17,50,18,1,3,17,2,2,13,18,16,2,25,5,5,
1,5,4,17,25,25,50,3,7,17,17,3,3,3,7,11,10,25,1,19,15,4,1,5,12,17,16,
50,20,20,20,25,50,2,17,3,20,20,20,5,1,18,15,2,3,25,12,9,3,3,19,16,20,
5,5,1,4,15,16,5,20,16,2,25,6,12,25,11,25,7,2,5,19,17,17,2,12)

prior_update <- c("25%" = 35, "50%" = 50, "75%" = 65)

for(i in 1:100){
  score_true <- x[i]
  S <- 4000
  s <- 1
  sigmas <- rep(NA, S)

  while(s <= S){
    sigma_get <- JQPDS_icdf(runif(1), lower_bound = 0, alpha = 0.25, quantiles = prior_update)
    x_get <- rnorm(1, 0, sigma_get)
    y_get <- rnorm(1, 0, sigma_get)
    polar_get <- car_to_polar(x_get, y_get)
    r_get <- polar_get[1]
    theta_get <- polar_get[2]
    score_get <- score(r_get, theta_get)
    if (score_true == score_get) {
      sigmas[s] <- sigma_get
      s <- s + 1
    }
  }
  prior_update <- quantile(sigmas, c(0.25, 0.50, 0.75))
}
```

```
summary(sigmas)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    23.62    27.78    29.39    29.85    31.47    44.69
```

```
hist(sigmas, breaks = 20)
```



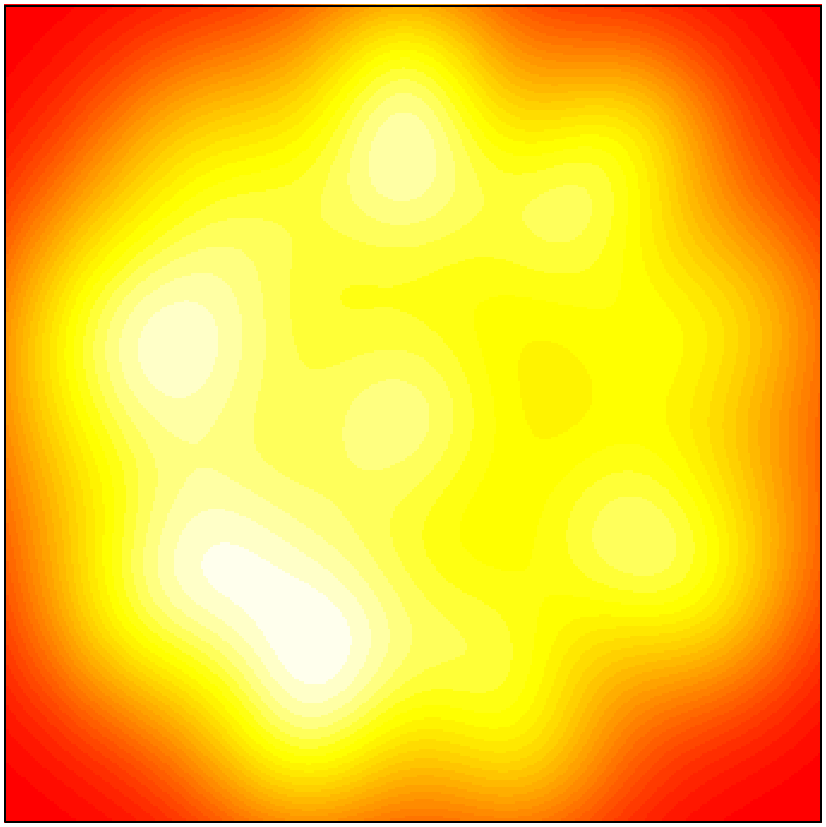
The posterior distribution has a median very close to the estimate produced by Tibshirani's implementation of the E-M algorithm, which is (as shown in exercise 1.4) 28.27. The advantage of this method over the `simpleEM` is that it does not only produce a point estimate, but an entire distribution of possible values for σ. The downside, however, is the excessive running time – although this could we partially solved by using more efficient methods to approximate the posterior distribution.

1.6 Optimization

A simple approach for this could be based in simulation. First, we will create a grid of potential targets in the board. For each target, we will simulate a large number of throws aiming at it with the given $\hat{\sigma}$. Then, we will average the scores in each of these points and select the target that produced the best scores.

Following Tibshirani's implementation and for the $\hat{\sigma}$ obtained in the previous exercise, we get:

```
exp_scores <- simpleExpScores(median(sigmas)^2)
drawHeatmap(exp_scores)
```



That means we should aim to the area surroinding the triple 19.

2 Manipulations of Continuous Probability Distributions

$U \sim U(0,1)$, and, for a given $k > 0$ and $w > 0$:

$$\theta = \frac{k}{(1-U)^{1/w}}$$

2.1 Parameter Space

$$\lim_{U \to 0} \frac{w}{(1-U)^{1/w}} = \frac{\kappa}{1^{1/w}} = k$$

$$\lim_{U \to 1} \frac{k}{(1-1)^{1/w}} = \frac{k}{0^{1/w}} = \infty$$

The parameter space for θ goes from κ to infinity. As $\kappa > 0$, we have $(0, \infty)$

2.2 Median

Considering that θ is a transformation of U and that the median for U is 0.5, the median for θ is:

$$Median(\theta) = \frac{k}{(0.5)^{1/w}}$$

2.3 Cumulative Density Function

The CDF, which corresponds to the inverse with respect to U, is:

$$F(\theta) = 1 - \left(\frac{k}{\theta}\right)^w$$

2.4 Probability Density Function

For the PDF we need to obtain partial derivate with respect to θ.

$$f(\theta) = \frac{\partial}{\partial\theta}\left(1 - \left(\frac{k}{\theta}\right)^w\right) = k^w w\theta^{-w-1}$$

2.5 Expectation

The expectation is:

$$\begin{aligned}
\mathbb{E}[\theta] &= \int_k^\infty \theta\frac{wk^w}{\theta^{w+1}}d\theta \\
&= \int_k^\infty \theta\frac{wk^w}{\theta^{w+1}}d\theta \\
&= wk^w\int_k^\infty \frac{\theta}{\theta^{w+1}}d\theta \\
&= wk^w\int_k^\infty \theta^{-w}d\theta \\
&= wk^w\left[\frac{\theta^{1-w}}{1-w}\bigg|_k^\infty\right] \\
&= wk^w\left[\frac{\infty^{1-w}}{1-w} - \frac{k^{1-w}}{1-w}\right]
\end{aligned}$$

The condition that has to be satisfied for $\mathbb{E}[\theta]$ to be finite is that $w > 1$. If that is the case, then:

$$\begin{aligned}
\mathbb{E}[\theta] &= wk^w\frac{-k^{1-w}}{1-w} \\
&= \frac{-wk}{1-w}
\end{aligned}$$

2.6 Prior Predictive Density Function

The marginal PDF of x is the integral of the joint probability of x and θ, which by definition corresponds to the product of the PDF of *theta* and the PDF of the conditional probability of x given *theta*.

$$f_X(x) = \int f(x,\theta)d\theta = \int f(\theta)f(x|\theta)d\theta$$

We have already seen that $f(\theta) = \kappa^\omega\omega\theta^{-\omega-1}$, which we reexpress as $f(\theta) = \frac{wk^w}{\theta^{w+1}}$ for convenience. If $X \sim U(0,\theta)$, then $f(x|\theta) = \frac{1}{\theta-0} = \frac{1}{\theta}$. Therefore:

$$\begin{aligned}
f_X(x) &= \int_k^\infty \frac{1}{\theta}\frac{wk^w}{\theta^{w+1}}d\theta \\
&= \int_k^\infty \frac{wk^w}{\theta^{w+2}}d\theta \\
&= \frac{-1}{w+1}\int_k^\infty \frac{wk^w}{\theta^{w+1}} \\
&= \left(\frac{-wk^w}{w+1}\right)\left[\frac{1}{\theta^{w+1}}\Bigg|_k^\infty\right] \\
&= \left(\frac{-wk^w}{w+1}\right)\left[\frac{1}{\infty^{(w+1)}}-\frac{1}{k^{(w+1)}}\right] \\
&= \frac{wk^w}{w+1}\frac{1}{k^{(w+1)}} \\
&= \frac{w}{(w+1)k}
\end{aligned}$$

2.7 Posterior Density Function

Given a particular observation x_i and following Bayes rule:

$$f(\theta|x_i) = \frac{f(\theta)f(x_i|\theta)}{f(x_i)} = \frac{f(\theta)f(x_i|\theta)}{\int f(\theta)f(x|\theta)d\theta}$$

We have already expressed the two components of the numerator and the denominator in the previous exercises. Thus:

$$\begin{aligned}
f(\theta|x_i) &= \frac{\frac{1}{\theta}\frac{wk^w}{\theta^{w+1}}}{\frac{w}{(w+1)k}} \\
&= \frac{1}{\theta^{w+2}}\frac{wk^w * k(w+1)}{w} \\
&= \frac{k^{w+1}(w+1)}{\theta^{(w+2)}}
\end{aligned}$$

2.8 Posterior Density Function Given N Observations

Given that we now consider N observations $\boldsymbol{x} = x_1, x_2, ..., x_N$:

$$f(\theta|\boldsymbol{x}) = \frac{f(\theta)f(\boldsymbol{x}|\theta)}{\int f(\theta)f(\boldsymbol{x}|\theta)d\theta}$$

Note that the expression is similar to the one in the previous exercise, except now we have a vector $\boldsymbol{x}$ instead of single observation x_i. This change does oviously not affect $f(\theta) = \frac{wk^w}{\theta^{w+1}}$, which does not depend on $\boldsymbol{x}$. However, it changes the likelihood and the marginal. Beginning with the likelihood, it can be expressed as:

$$f(\boldsymbol{x}|\theta) = \prod_{i=1}^{N} f(x_i|\theta) = \prod_{i=1}^{N}\frac{1}{\theta} = \frac{1}{\theta^N}$$

Considering this result, the marginal is:

$$f(\boldsymbol{x}) = \int_k^\infty \frac{1}{\theta^n}\frac{wk^w}{\theta^{w+1}}d\theta = \frac{w}{w+n}\frac{1}{k^n}$$

Therefore, the posterior PDF is:

$$\begin{aligned}
f(\theta|\boldsymbol{x}) &= \frac{f(\theta)f(\boldsymbol{x}|\theta)}{\int f(\theta)f(\boldsymbol{x}|\theta)d\theta} \\
&= \frac{\frac{1}{\theta^N}\frac{wk^w}{\theta^{w+1}}}{\frac{w}{w+n}\frac{1}{k^n}} \\
&= \frac{1}{\theta^{n+w+1}}wk^w(\frac{k^n(w+1)}{w}) \\
&= \frac{k^{w+n}(w+1)}{\theta^{(w+1+n)}}
\end{aligned}$$

2.9 Posterior Predictive Density Function

For a new $\tilde{x}$, the posterior preditive distribution is defined as:

$$f(\tilde{x}|\boldsymbol{x},\theta) = \int p(\tilde{x}|\theta)p(\theta|\boldsymbol{x})$$

Considering that $p(\tilde{x}|\theta) = \frac{1}{\theta}$ and the $p(\theta|\boldsymbol{x})$ that we calculated in the previous section, we have:

$$\begin{aligned}
f(\tilde{x}|\boldsymbol{x}) &= \int_k^\infty \frac{1}{\theta}\frac{k^{w+n}(w+1)}{\theta^{(w+1+n)}}d\theta \\
&= k^{w+n}(w+1)\int_k^\infty \theta^{-(w+2+n)}d\theta \\
&= k^{w+n}(w+1)\left[\frac{-\theta^{-(w+n+1)}}{w+n+1}\Bigg|_k^\infty\right] \\
&= \frac{k^{n+w}(w+1)k^{-n-w-1}}{w+n+1} \\
&= \frac{w+1}{(w+n+1)k}
\end{aligned}$$