(SMM641) - Revenue Management & Pricing. Quantity Based Revenue Management Part 1 - R Supplement

Oben Ceryan

24 January 2022

Contents

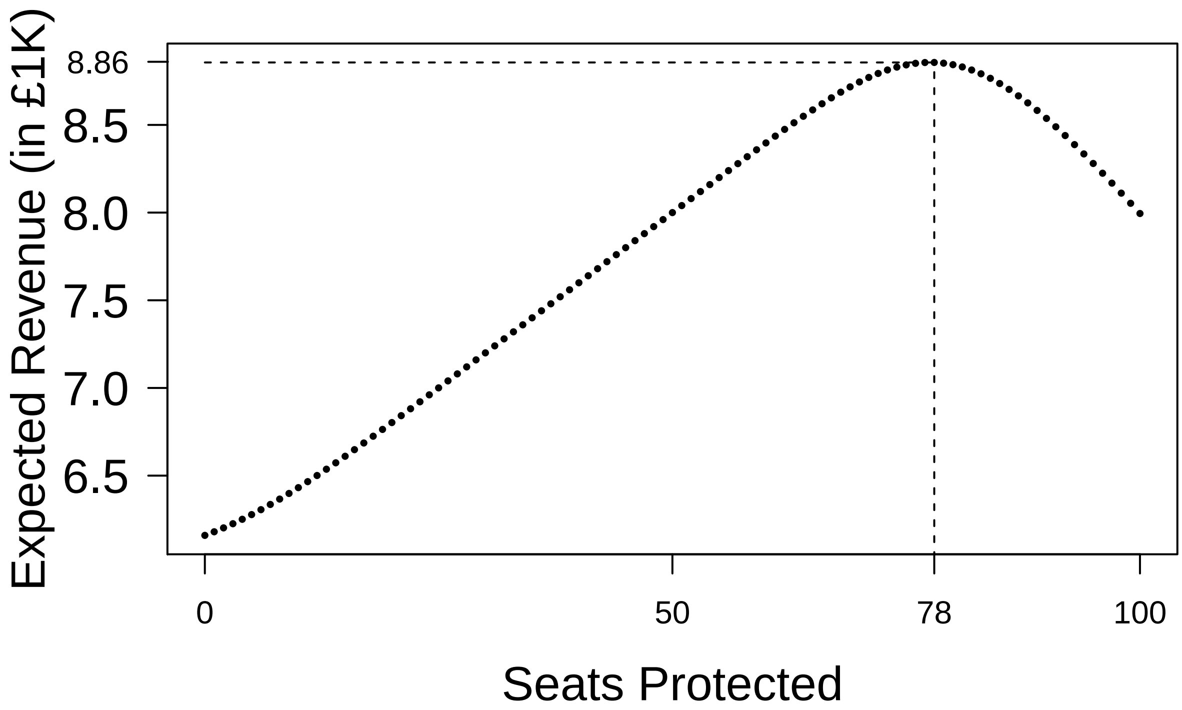
1 Single Resource Revenue Management with Two Fare Classes

1.1 Optimal Protection Level for High-Fare Demand: Numerical Inspection

```r
# A NUMERICAL EXAMPLE USING POISSON DISTRIBUTION
mL=100          # Mean Demand for Low-Fare, Poisson
mH=80           # Mean Demand for High-Fare, Poisson
pL=60           # Price for Low-Fare
pH=100          # Price for Low-Fare
capacity=100    # Capacity
ExpRevenue=rep(0,capacity+1)
for (i in 1:(capacity+1)){
    protect=i-1
    availforLowFare=capacity-protect;
    ExpRevenue[i]=0;
    for(dL in 0:200){
        soldLowFare=min(availforLowFare,dL)
        remainforHighFare=capacity-soldLowFare
        for(dH in 0:200){
            soldHighFare=min(remainforHighFare,dH)
            RevenueThisIter=pL*soldLowFare+pH*soldHighFare
            ExpRevenue[i]=ExpRevenue[i]+
                RevenueThisIter*dpois(dL,mL)*dpois(dH,mH)
        }
    }
}
Protectindexbest = which(ExpRevenue == max(ExpRevenue))
ProtectBest=Protectindexbest-1
OptimalExpRevenue=max(ExpRevenue)
print(paste("The Optimal Protection Level for High-Fare Demand:", ProtectBest))
```

```
## [1] "The Optimal Protection Level for High-Fare Demand: 78"
```

```
# Plotting Expected Revenue vs Protection Level
xaxis=0:capacity
plot(xaxis,ExpRevenue/1000,pch = 16, cex = 0.5,las=1, xaxt="n",
     xlab="Seats Protected",ylab="Expected Revenue (in £1K)",cex.lab=1.5, cex.axis=1.5, cex.ma
xticks <- seq(0, capacity, by=50)
axis(side = 1, at = xticks)
axis(side = 1, at = ProtectBest)
lines(c(ProtectBest,ProtectBest),c(0, max(ExpRevenue)/1000),lty=2)
axis(side = 2, at = round(max(ExpRevenue)/1000,2),las=1)
lines(c(0,ProtectBest),c(max(ExpRevenue)/1000, max(ExpRevenue)/1000),lty=2)
```



1.2 Optimal Protection Level for High-Fare Demand: Analytical Solution

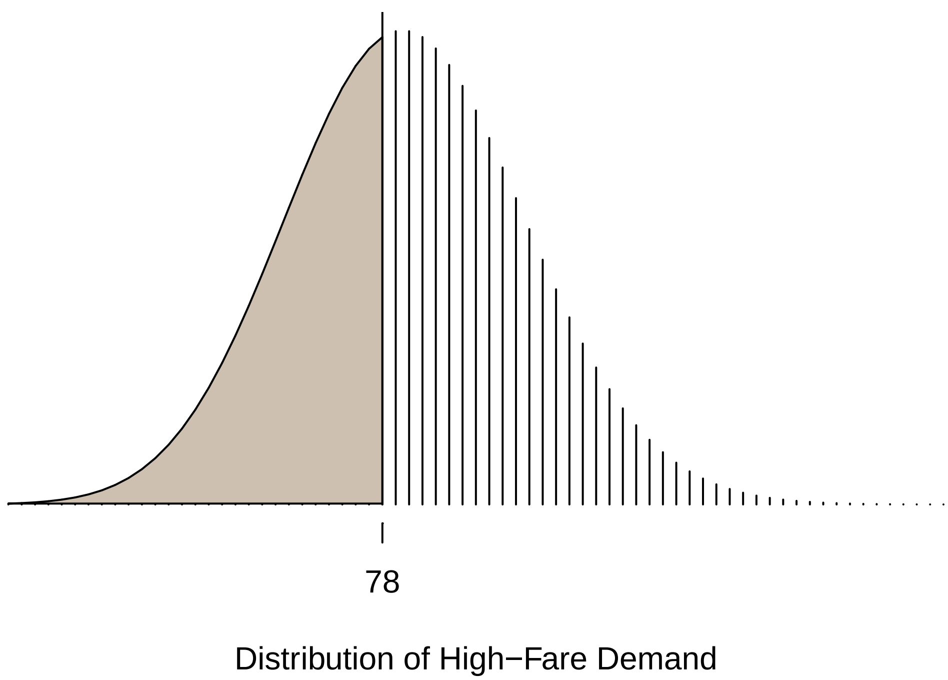
Critical Fractile $= \frac{p_H - p_L}{p_H} = \frac{100-60}{100} = \frac{40}{100} = 0.4.$

Set protection level, y^*, such that $P(D \leq y^*) = 0.4.$

```r
# ANALYTICAL EXPRESSION
critFrac=(pH-pL)/pH
protect<-qpois(critFrac, mH)

# Plotting:
plot.new()
x <- seq(50, 120, length=71)
y <- dpois(x, mH)
plot(x,y,type="h",xaxt="n",yaxt="n", bty="n",
     xlab = "Distribution of High-Fare Demand", ylab = "")

polygon(c(x[x<=qpois(critFrac, mH)],qpois(critFrac, mH)),
        c(y[x<=qpois(critFrac, mH)],y[x==50]),
        col="antiquewhite3")
lines(c(protect,protect),c(0, qpois(critFrac, mH)))
axis(side = 1, at = round(protect))
```



1.3 Optimal Booking Limit for Low-Fare Demand

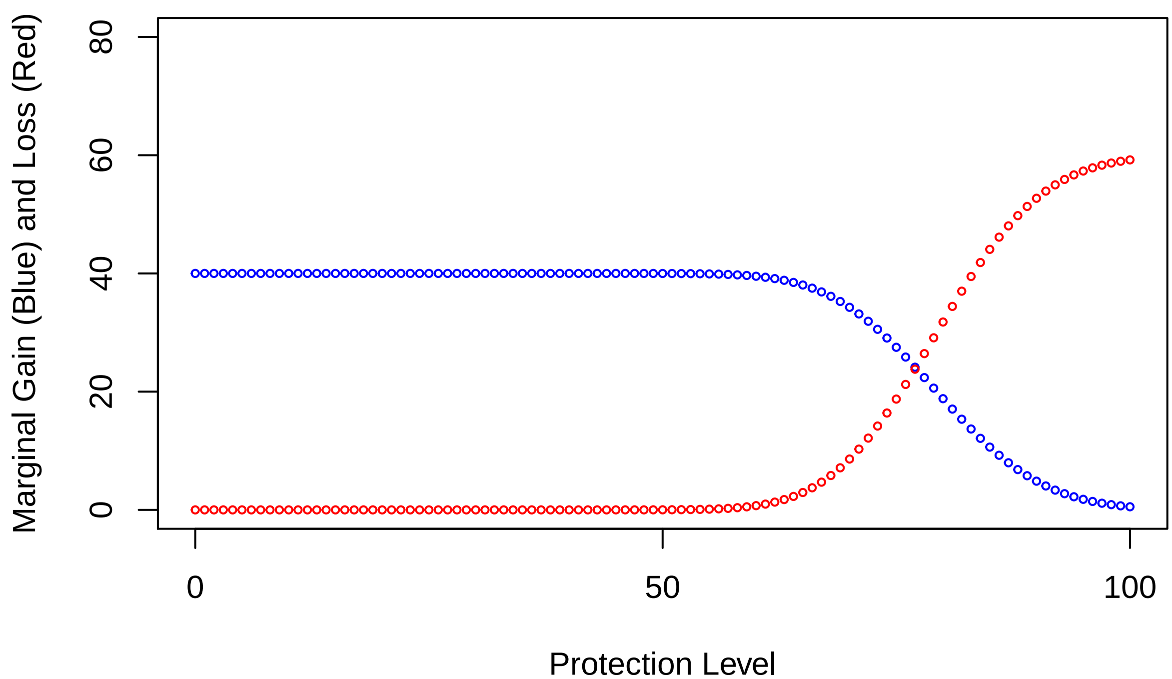
Booking Limit for Low-fare Demand = Capacity - Protection Level for High-fare Demand.

```
bookingLimit=capacity-ProtectBest
print(bookingLimit)
```

```
## [1] 22
```

1.4 Marginal Gain vs Marginal Loss

```
ExpGain=rep(0,capacity+1)
ExpLoss=rep(0,capacity+1)
for (i in 1:(capacity+1)){
    protect=i-1
    ExpGain[i]=(1-ppois(protect,mH))*(pH-pL)
    ExpLoss[i]=ppois(protect,mH)*pL
}
xaxis=0:capacity
plot(xaxis,ExpGain,type="p",cex=0.5,col="blue",ylim=c(0,80),xaxt="n",
     xlab = "Protection Level", ylab = "Marginal Gain (Blue) and Loss (Red)")
lines(xaxis,ExpLoss,type="p",cex=0.5,col="red")
xtick<-seq(0, 100, by=50)
axis(side=1, at=xtick)
```



1.5 Optimal Protection Level with Continuous Demand Distributions

Suppose high-fare demand is normal with mean 80 and standard deviation 8.94, i.e., $\sqrt{80}$.

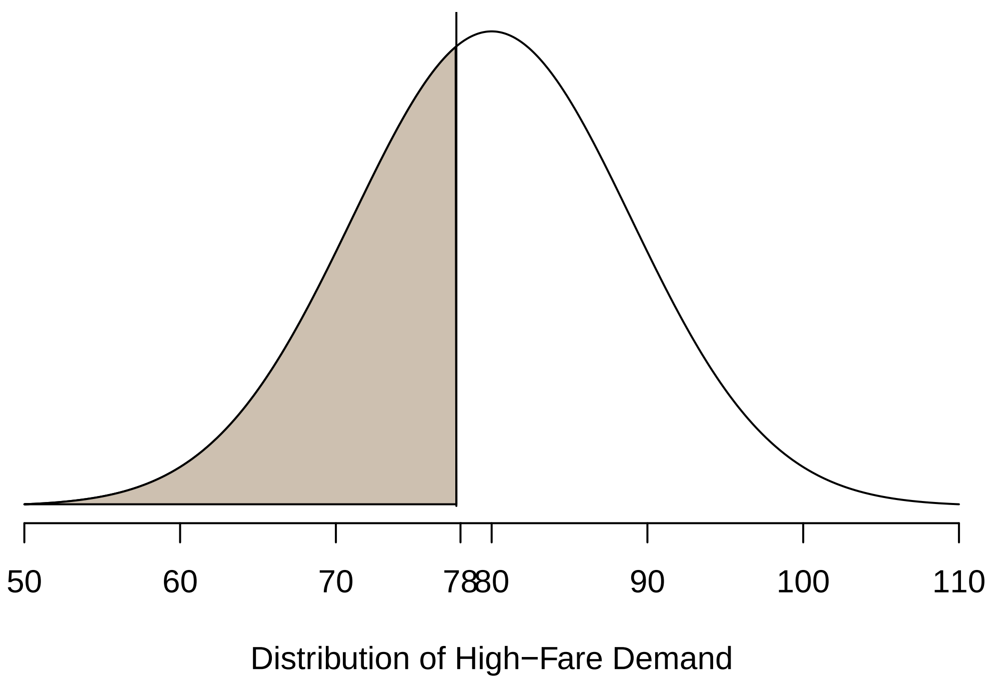
Critical Fractile $= \frac{p_H - p_L}{p_H} = \frac{100-60}{100} = \frac{40}{100} = 0.4$.

Set protection level, y^*, such that $P(D \leq y^*) = 0.4$.

```r
mH=80;
sH=sqrt(mH);
critFrac=(pH-pL)/pH
protect<-qnorm(critFrac, mH,sH)

# Plotting:
plot.new()
x <- seq(50, 110, length=1000)
y <- dnorm(x, mH,sH)
plot(x,y,type="l",yaxt="n", bty="n",
     xlab = "Distribution of High-Fare Demand", ylab = "")

polygon(c(x[x<=qnorm(critFrac, mH,sH)],qnorm(critFrac, mH,sH)),
        c(y[x<=qnorm(critFrac, mH,sH)],y[x==50]),
        col="antiquewhite3")
lines(c(protect,protect),c(0, qnorm(critFrac, mH,sH)))
axis(side = 1, at = round(protect))
```



1.6 A Lower Bound for Expected Revenue (FCFS)

Suppose we admit demand on a first come first serve (FCFS) basis, i.e., set protection level to zero.

```
mL=100          # Mean Demand for Low-Fare, Poisson
mH=80           # Mean Demand for High-Fare, Poisson
pL=60           # Price for Low-Fare
pH=100          # Price for Low-Fare
capacity=100    # Capacity
ExpRevenue=rep(0,capacity+1)
for (i in 1:1){
    protect=i-1
    availforLowFare=capacity-protect;
    ExpRevenue[i]=0;
    for(dL in 0:200){
        soldLowFare=min(availforLowFare,dL)
        remainforHighFare=capacity-soldLowFare
        for(dH in 0:200){
            soldHighFare=min(remainforHighFare,dH)
            RevenueThisIter=pL*soldLowFare+pH*soldHighFare
            ExpRevenue[i]=ExpRevenue[i]+RevenueThisIter*dpois(dL,mL)*dpois(dH,mH)
        }
    }
}
RevenueFCFS=ExpRevenue[1]
print(paste("Lower Bound for Expected Revenue (FCFS):", round(RevenueFCFS,1)))
```

```
## [1] "Lower Bound for Expected Revenue (FCFS): 6159.4"
```

1.7 An Upper Bound for Expected Revenue (Perfect Foresight)

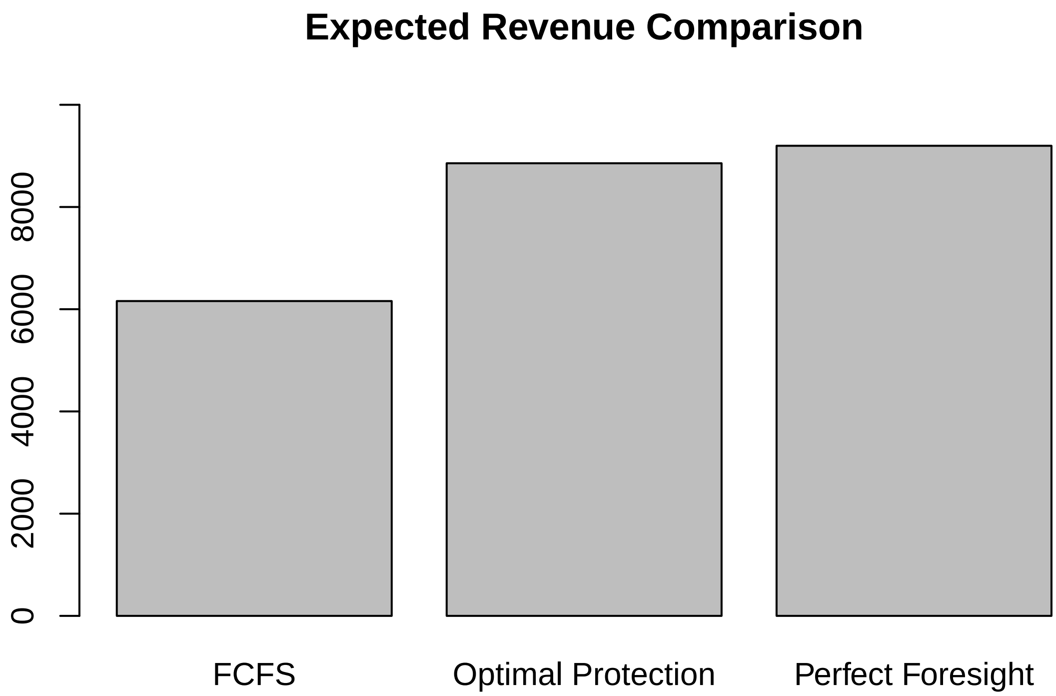
Suppose we have perfect foresight, i.e., can see future demand and apply best allocation

```
mL=100          # Mean Demand for Low-Fare, Poisson
mH=80           # Mean Demand for High-Fare, Poisson
pL=60           # Price for Low-Fare
pH=100          # Price for Low-Fare
capacity=100    # Capacity
ExpRevenueUB=0
for(dL in 0:200){
for(dH in 0:200){
    soldHighFare=min(dH,capacity)
    remainforLowFare=capacity-soldHighFare
    soldLowFare=min(dL,remainforLowFare)
    RevenueThisIter=pL*soldLowFare+pH*soldHighFare
    ExpRevenueUB=ExpRevenueUB+RevenueThisIter*dpois(dL,mL)*dpois(dH,mH)
}
}
print(paste("Upper Bound for Expected Revenue (Perfect Foresight):", round(ExpRevenueUB,1)))
```

```
## [1] "Upper Bound for Expected Revenue (Perfect Foresight): 9197.9"
```

1.8 Revenue Potential of the Optimal Protection Level

```r
barplot(c(RevenueFCFS,OptimalExpRevenue,ExpRevenueUB),
        names.arg = c("FCFS","Optimal Protection","Perfect Foresight"),
        main="Expected Revenue Comparison",
        ylim=range(pretty(c(0, ExpRevenueUB))))
```




```r
print(paste("Optimal protection achieves:",
            round((OptimalExpRevenue-RevenueFCFS)/(ExpRevenueUB-RevenueFCFS),2),
            " of revenue potential of perfect foresight."))
```

```
## [1] "Optimal protection achieves: 0.89  of revenue potential of perfect foresight."
```