Module 3

This is a single, concatenated file, suitable for printing or saving as a PDF for offline viewing. Please note that some animations or images may not work.

Module 3 Study Guide and Deliverables

Topics:	**Lecture 5:** Analyzing Risk: Modeling Input Data **Lecture 6:** Analyzing Risk: Dealing with Correlated Data
Readings:	Lectures 5 and 6 online content
Discussions:	No discussion.
Assignments:	**Tutorial:** 3 **Assignment 3**: Individual Assignment covering Lecture 1 and Lecture 6.
Assessments:	**Quiz 3**

Lecture 6

Learning Objectives

After you complete this lecture, you will be familiar with the following:

- Pearson product moment correlation
- Spearman rank correlation
- Computing Pearson product moment correlation and Spearman rank correlation in R

- Incorporating a correlation matrix into an R model
- Impact of correlation on the decision-making process

Zappos' Problem

In the first part of this module, we will be working with the following problem:

Zappos is a retailer of women's clothing, accessories, and beauty products. Suppose that the regional purchasing manager for Zappos is in the process of determining the order quantities for a new product line of women's sleepwear for the upcoming holiday season. The new product line consists of four types of pajamas: cotton, flannel, silk, and velour. An initial order quantity of 10,000 of each type of pajama has been proposed, but we want to evaluate this decision with a simulation model. We are given the following data for this problem.

	Cost	Selling Price	Discount Price
Cotton	$25	$30	$10
Flannel	$25	$40	$10
Silk	$35	$60	$30
Velour	$30	$55	$20

The cotton and the flannel pajamas are similar to other products that Zappos has sold in the past. The marketing department has compiled representative samples of cotton and flannel pajama demand as shown in Figure 6.1.

The silk pajama's design is different from that of all other Zappos' existing products, therefore the marketing department does not believe that we should use past data to model the demand of silk pajamas. However, the marketing department has conducted extensive surveys to estimate the likely demand scenarios for silk pajamas and obtained the following information:

Values	Demand
5,000	0.1
10,000	0.4
15,000	0.2
20,000	0.25
25,000	0.05

In addition, Zappos estimates a maximum silk pajama demand of 30,000 and a minimum possible demand of 0.

The velour pajama is a new product being introduced this season. Therefore, we cannot use past data to model it and the only information that we have is the minimum, most likely, and maximum values of velour pajama demand, which are 2500 units, 10,000 units, and 25,000 units, respectively.

Source: Camm, J. D., Cochran, J. J., Fry, M. J., Ohlmann, J. W., Anderson, D. R., Sweeney, D. J., & Williams, T. A. (2015). *Essentials of business analytics* (1st edition). Cengage Learning, pp. 506–514.

Figure 6.1

Period	Cotton	Flannel
1	21,311	6,800
2	13,311	15,052
3	17,573	9,145
4	17,994	12,601
5	21,710	9,978
6	22,882	7,052
7	17,279	16,298
8	25,218	3,235
9	19,315	12,241
10	22,553	9,563
11	23,841	3,454
12	18,969	12,327
13	23,058	9,651
14	22,497	9,494
15	17,334	15,241
16	21,659	9,684
17	21,443	3,401
18	20,018	19,839
19	20,409	5,252
20	15,571	14,290
21	18,333	12,006
22	25,657	6,580
23	16,013	12,848
24	14,191	20,398

Input Modeling for Zappos Problem

Modeling the cotton pajama demand

The historical data for cotton pajama demand is given in Figure 6.1. We now use the fitdistrplus package to find a distribution that fits our cotton pajamas data nicely. After trying several distributions, we might settle on the normal distribution. We use the fitdist function to create a normal model from our data and the gofstat function to display goodness of fit statistics about it. Both of these are from the “fitdistrplus” library. The code for this is reproduced in Figure 6.2

Figure 6.2

```
> HData<-read.csv("~/L6 Zappos Data.csv")
>
> cModel<-fitdist(HData$Cotton,"norm")
> gofstat(cModel)
Goodness-of-fit statistics
                               1-mle-norm
Kolmogorov-Smirnov statistic 0.12278034
Cramer-von Mises statistic   0.03669431
Anderson-Darling statistic   0.21860919

Goodness-of-fit criteria
                                1-mle-norm
Akaike's Information Criterion   460.5512
Bayesian Information Criterion   462.9073
> cModel$estimate
     mean        sd
19922.458  3269.999
>
```

The demand for pajamas is indeed a discrete random variable. However, a typical practice to model the demand is to use a continuous distribution (as discussed in Lecture 5). As the numbers a discrete random variable can take on get larger, continuous distributions become stronger approximations.

We observe a KS statistic of .12, which, while not perfect, is OK for our purposes. Since our sample is relatively small, random noise in the data makes it very unlikely to find any strong fit for our data.

We also want to make sure that “out of bounds” values, in this case, demand values less than 0, won’t pop up in our simulations. We see the mean is approximately six standard deviations higher than 0. We’ll almost never encounter a trial with demand lower than 0 under such circumstances.

Modeling the flannel pajama demand

After trying out different distributions, we eventually settle on the Weibull distribution to model flannel pajama demand.

Individual Exercise:

Try “shifting” the distribution by subtracting or adding values to the data set before fitting it. Can you get a better fit than we use in this lecture?

We again produce a model, along with parameter estimation and fit statistics, reproduced in Figure 6.3.

Figure 6.3

```
> fModel<-fitdist(HData$Flannel,"weibull")
> gofstat(fModel)
Goodness-of-fit statistics
                                1-mle-weibull
Kolmogorov-Smirnov statistic     0.10422565
Cramer-von Mises statistic       0.03777254
Anderson-Darling statistic       0.27424385

Goodness-of-fit criteria
                                1-mle-weibull
Akaike's Information Criterion       476.3109
Bayesian Information Criterion       478.6670
> fModel$estimate
       shape        scale
    2.475211 12061.534760
>
```

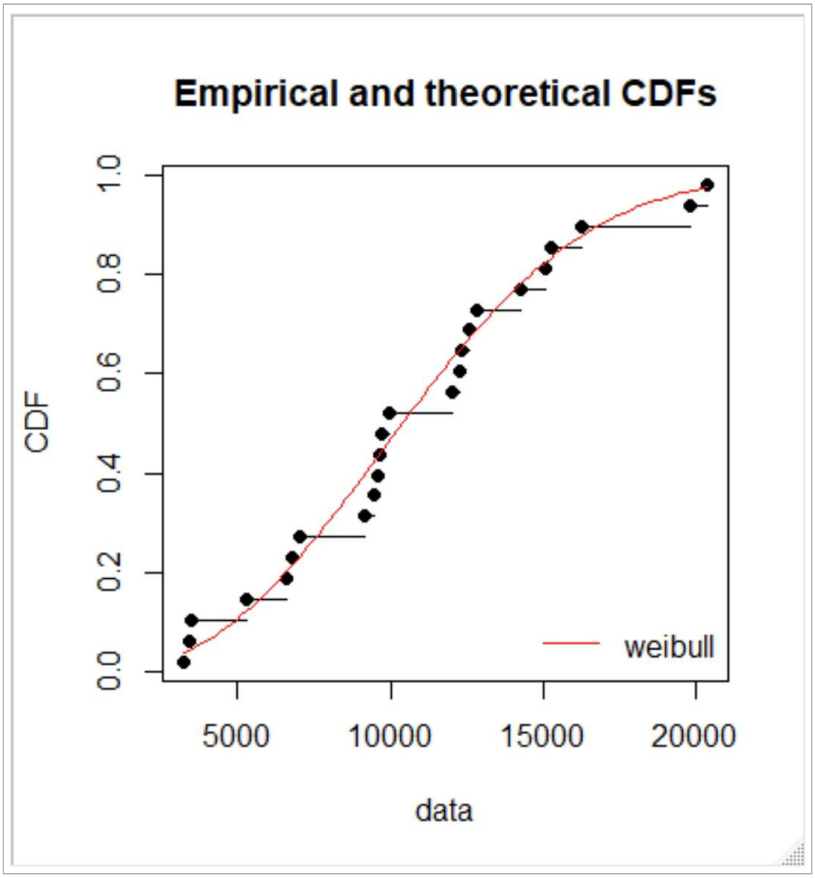
The fitdistrplus library makes it very easy to do graphical fit tests of our distributions as well. In figure 6.4, we compare the cdf and pdf of our fitted distribution to that of our data, and produce qq and pp plots as well.

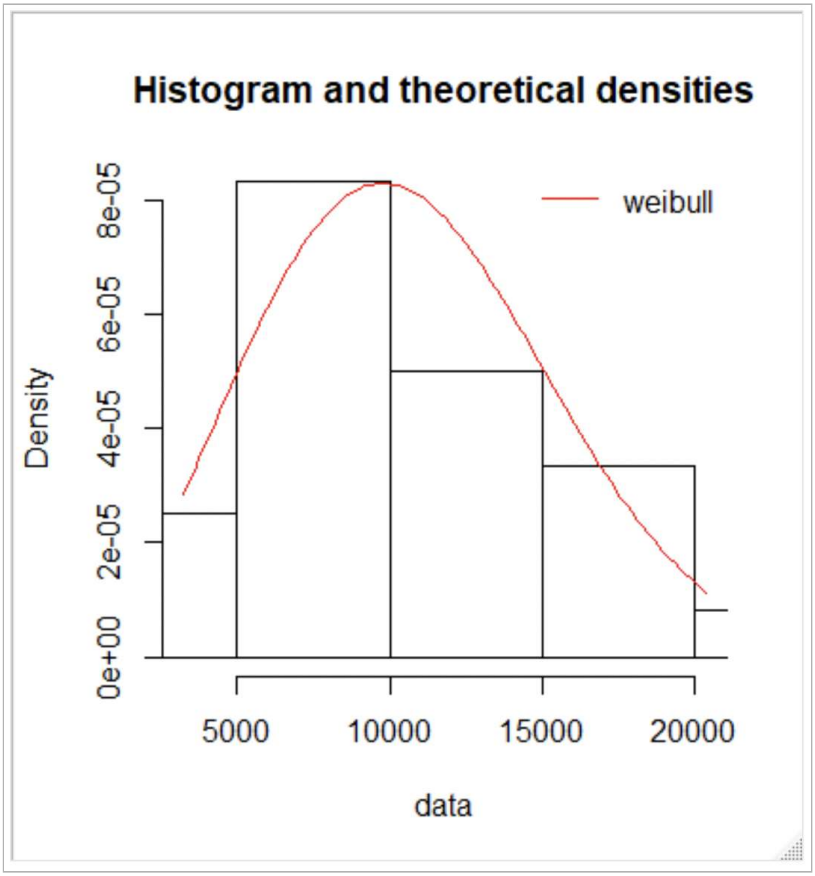
Individual Exercise:

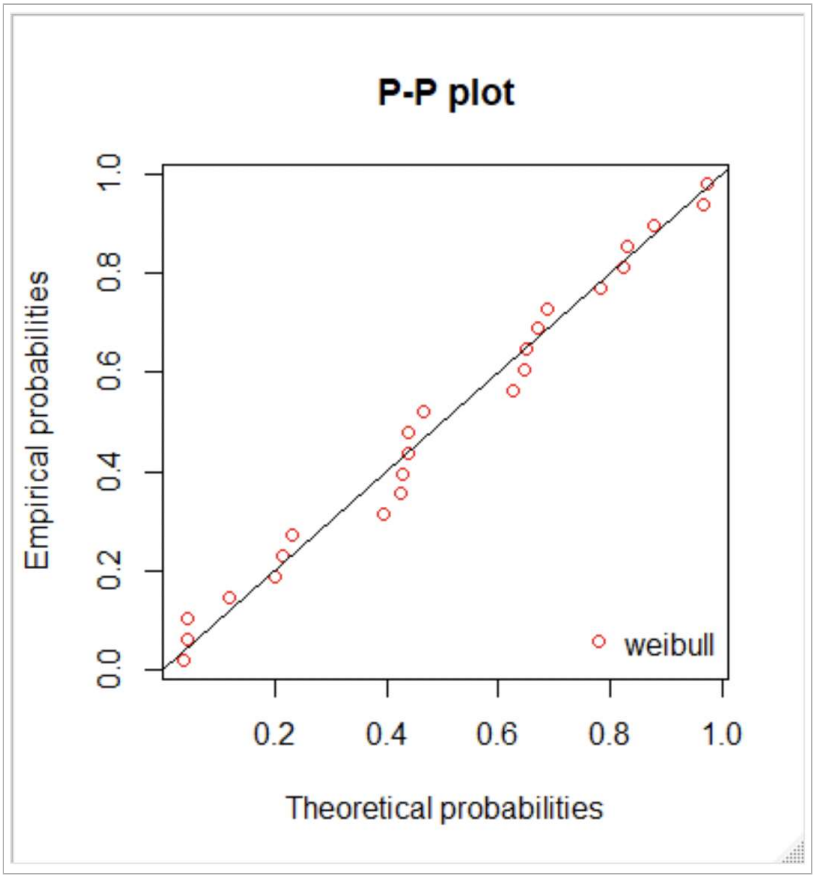
Check the examples provided in the documentation for fitdistrplus to see how to compare multiple potential fits at once.

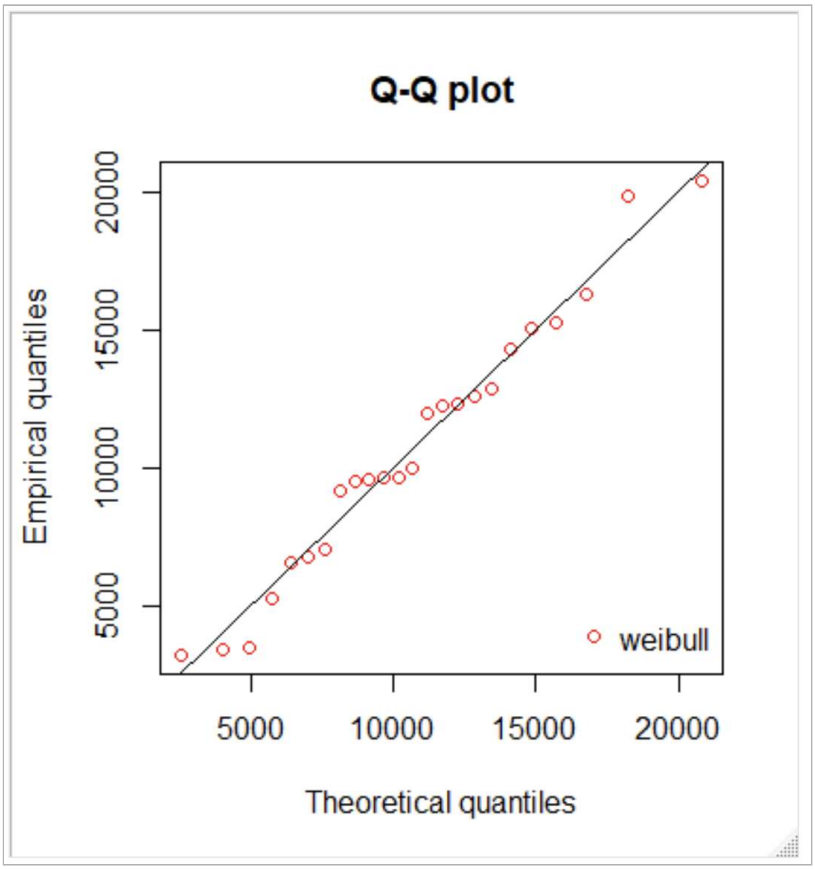
Figure 6.4

```
cdfcomp(fModel)
denscomp(fModel)
qqcomp(fModel)
ppcomp(fModel)
```


Empirical and theoretical CDFs
CDF
0.0
0.2
0.4
0.6
0.8
1.0
5000
10000
15000
20000
data
weibull


Histogram and theoretical densities
Density
0e+00
2e-05
4e-05
6e-05
8e-05
5000
10000
15000
20000
data
weibull


P-P plot
Empirical probabilities
0.0
0.2
0.4
0.6
0.8
1.0
weibull
0.2
0.4
0.6
0.8
1.0
Theoretical probabilities


Q-Q plot
Empirical quantiles
5000
10000
15000
20000
weibull
5000
10000
15000
20000
Theoretical quantiles

The demand for silk pajamas is provided directly by the problem. We use the breakpoints method defined in the previous lecture to handle this. The code for this can be seen in Figure 6.5.

Figure 6.5

```
x<-runif(n)
s<-sapply(X=x,
          function(t){
            if(t<.1) rdunif(n=1,a=1,b=5000)
            else if (t<.5) rdunif(n=1,a=5001,b=10000)
            else if (t<.7) rdunif(n=1,a=10001,b=15000)
            else if (t<.95) rdunif(n=1,a=15001,b=20000)
            else rdunif(n=1,a=20001,b=25000)
          })
```

Modeling velour pajama demand

We use a triangular distribution to model demand for velour pajamas, as the problem only specifies a minimum, maximum, and most likely value. We combine all four demands into one data frame in Figure 6.6, reordering them alphabetically.

Figure 6.6

```
n<-10000 #create a dataframe simulating demand
df<-data.frame(demC=rnorm(n,cModel$estimate[1],cModel$estimate[2]) %>% round(),
               demF=rweibull(n,fModel$estimate[1],fModel$estimate[2]) %>%round(),
               demV=rtri(n,2500,10000,25000) %>%round()
               )
df$demS<-s
rm(x,s)
df<-select(df,dcotton,dflannel,dsilk,dvelour)
```

Model Output

We now study the output of the model by computing the profit, using the table from the statement of the problem to determine costs and revenues. The code for this, as well as a histogram representing profits, and the mean and standard deviation, are provided in Figure 6.7.

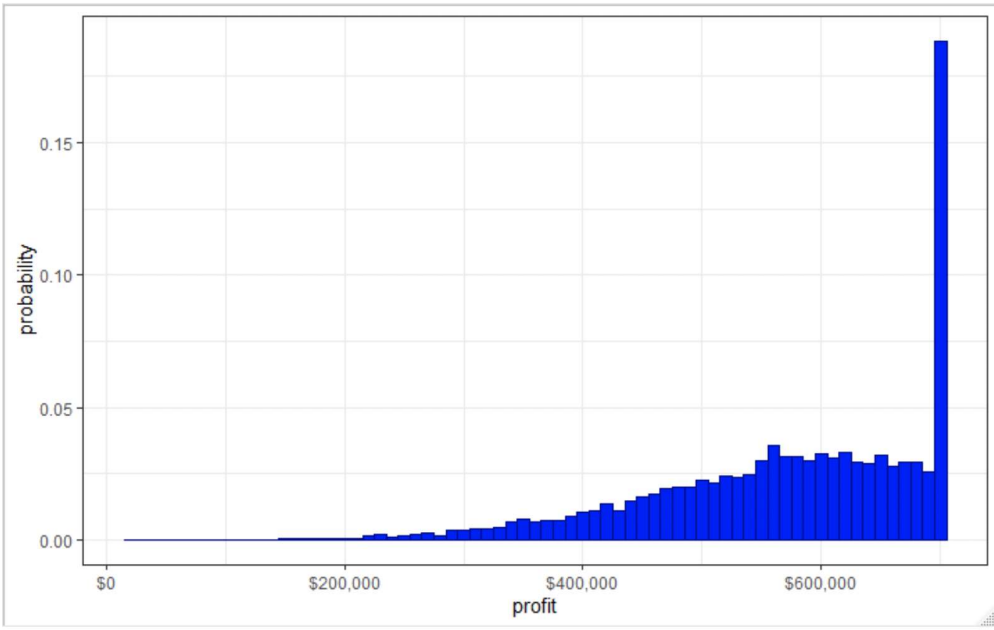
Figure 6.7

```
orderqty<-10000 #order quantity, unit revenues (full price and discounted), and un
revC<-30
revF<-40
revS<-60
revV<-55
discC<-10
discF<-10
discS<-30
discV<-20
costC<-25
costF<-25
costS<-35
costV<-30

profit<-transmute(df, profit=revC*pmin(orderqty,demC)+discC*pmax(orderqty-demC,0)+
                  revF*pmin(orderqty,demF)+discF*pmax(orderqty-demF,0)+
                  revS*pmin(orderqty,demS)+discS*pmax(orderqty-demS,0)+
                  revV*pmin(orderqty,demV)+discV*pmax(orderqty-demV,0)-
                  orderqty*(costC+costF+costS+costV))
profit<-profit$profit #we used a transmute before, now we overwrite our dataframe v

cat("mean profit:",mean(profit),"\nsd profit:",sd(profit),"\n")

ggplot()+geom_histogram(aes(x=profit,y=..count../sum(..count..)),
                        color="dark blue",fill="blue",binwidth=10000)+
  theme_bw()+ylab("probability")+
  scale_x_continuous(labels=scales::dollar)
```



Correlation Among the Inputs

- Suppose the market research has suggested that the cotton and flannel pajamas are often substitutes. In other words, customers buy either

one or another. Therefore, if the demand for one of them is high, we expect the demand for the other to be low.

- Research also suggests that the demand of silk and velour pajamas is negatively correlated with an estimated correlation of –0.5.
- Cotton pajama demand is positively correlated with both silk and velour pajamas with a correlation of 0.25.
- Correlation between flannel and velour pajamas is estimated to be 0.25.
- Correlation between flannel and silk pajamas is estimated to be 0.

Source: Adapted from Camm, J. D., Cochran, J. J., Fry, M. J., Ohlmann, J. W., Anderson, D. R., Sweeney, D. J., & Williams, T. A. (2015). *Essentials of business analytics* (1st edition). Cengage Learning, pp. 538, 540.

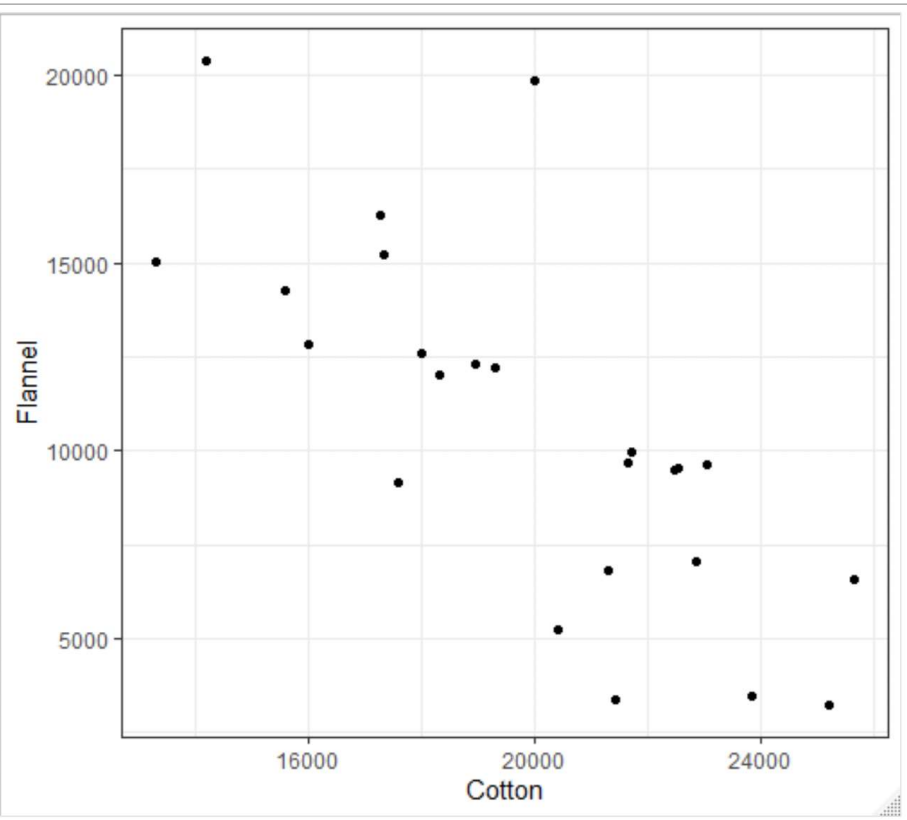
Our goal in this section is to incorporate the given correlations into our model. First, let us review the basics for correlation.

Pearson Product Moment Correlation

Pearson product moment correlation is the most widely used correlation measure. It measures the strength of linear relationship between two random variables. It takes values between –1 and 1. A Pearson product moment correlation of –1 denotes a strong negative relationship between a pair of random variables, while a Pearson product moment correlation of +1 denotes a strong positive relationship between a pair of random variables. A Pearson product moment correlation close to 0 indicates the absence of linear relationship between a pair of random variables.

We are given the information that the cotton and flannel pajamas are often substitutes but we are not given any information about the degree of their correlation. However, given the historical data of cotton and flannel pajamas, we can visually check their relationship to one another. We do so in Figure 6.8, noting what looks to be a strong negative correlation. If we use the cor function, our suspicions will be confirmed: we observe a correlation of –.75. This information is missing from our previous model. We'll now see how to incorporate it.

Figure 6.8




```
ggplot(HData)+geom_point(aes(x=Cotton,y=Flannel)) +theme_bw()
```

Spearman Rank Correlation

Eventually, we're going to be simulating demands for cotton and flannel pajamas, following the distributions we determined in the previous section. However, since they follow different distributions, and since Pearson correlation measures the strength of the linear relationship between them, the Pearson correlation can only go so high. To account for this, we use the Spearman correlation instead.

The Spearman correlation takes the intermediate step of calculating the rank of each observation in the data frame, then computing the Pearson correlation of those ranks. If these ranks match up perfectly, you could observe a Spearman correlation of 1, when the Pearson correlation would not achieve that result. In other words, Spearman correlation allows us to generate correlated values from different probability distributions.

We compute the Spearman correlation using the "method" argument within the cor function, as displayed in Figure 6.9.

Figure 6.9

```
> cor(HData$Cotton,HData$Flannel,method="spearman")
[1] -0.7713043
```

Setting Up a Correlation Matrix in ASP

To incorporate correlations in our simulation, we'll use the NORTARA library. First, however, we'll define the correlation matrix our data will follow. A correlation matrix between n random variables simply defines the pairwise correlation of each pair of random variables. Since ρ_XY=ρ_YX, each element of the matrix should be the same as that element with the indices flipped. For example, if the element in the second column of the first row is 0.6, the element in the first column of the second row should also be 0.6. Also, since the correlation of any random variable with itself is 1, the top right to bottom left diagonal should contain only 1's. Using the inputs provided by the problem, we set up our correlation matrix as in Figure 6.10:

Figure 6.10

```
cormatrix<-matrix(c(   1,-.77,  .25, .25,
                    -.77,   1,    0, .25,
                     .25,   0,    1, -.5,
                     .25, .25,  -.5,   1)
                  ,4,4)
```

The matrix function takes a vector of numbers, a number of rows, and a number of columns, and returns a matrix of those dimensions. If we're careful spacing the vector, it makes it very easy to see what element will go where in our matrix.

Unfortunately, even if we follow the specifications listed above, we may not have a valid correlation matrix. For example, if a matrix were as follows:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

This would mean that the first and second elements have a perfect linear relationship, and the second and third elements also have a perfect linear relationship, but the first and third are completely uncorrelated. This is impossible; the first and third elements would need to also be related linearly. To test to make sure our matrix of correlations is "valid" in these terms, we need to know that it is positive semidefinite. We can use the is.positive.semi.definite function from the matrix.calc library to make this determination. If it is not, we can adjust

it using the nearPD function from the Matrix library, which adjusts the values in the matrix as little as possible to get a positive semidefinite matrix. The code and results can be seen in Figure 6.11.

Figure 6.11

```
> is.positive.semi.definite(cormatrix)
[1] FALSE
> cormatrix2<-matrix(nearPD(cormatrix,corr=TRUE)$mat,4,4)
> cormatrix2
           [,1]        [,2]        [,3]       [,4]
[1,]  1.0000000 -0.75004183  0.23718680  0.2339270
[2,] -0.7500418  1.00000000 -0.01174761  0.2352636
[3,]  0.2371868 -0.01174761  1.00000000 -0.4905392
[4,]  0.2339270  0.23526364 -0.49053921  1.0000000
```

Building the Correlated Model

Unfortunately, we can't just build a data frame like we did before; we now need to account for correlations. To do this, we need the draw.d.variate.uniform() function in the "MultiRNG" package, but as usual there's a little prep work we have to do. In addition to a number of trials and a correlation matrix, the function requires us to provide as inputs the inverse cdf functions of our correlated variables, along with the parameters for those variables.

For cotton and flannel pajama demand, this is no problem; we have the qnorm and qweibull functions from R. For the velour pajamas, we've already written an rtri function, so we just use the same function without the sapply to create qtri.

For the silk pajamas, we still need to come up with the inverse cdf function. Without getting into too many details, we start with the density function, integrate, then take the inverse of the result. Fortunately, the density function of silk pajamas was piece-wise constant, so this is fairly easy to do.

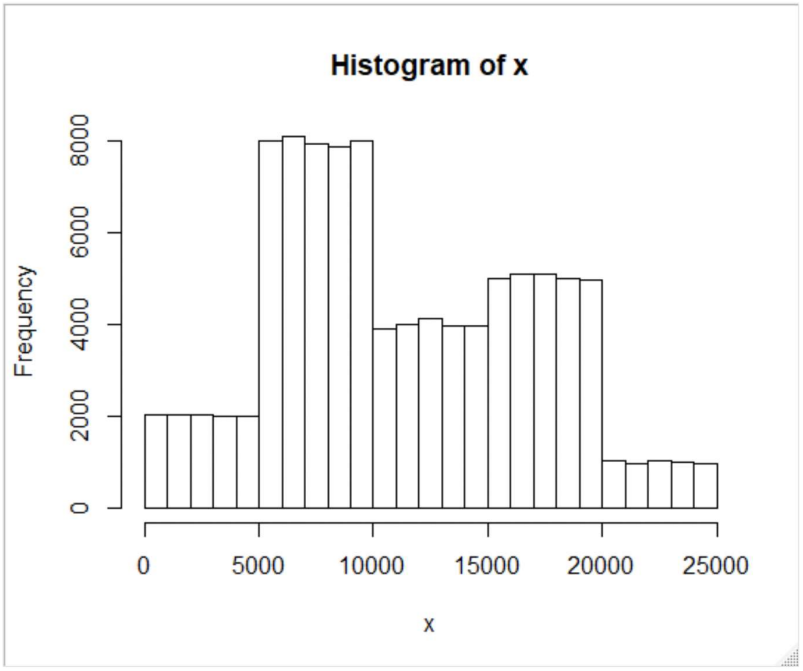
The inverse cdfs for the triangular distribution and our specific silk pajama density are provided in Figure 6.12, along with a histogram of randomly generated silk pajamas to make sure the function works as intended.

Figure 6.12

```
qtri<-function(U,min,ml,max){ #we need inverse cdfs to feed ger
  F<-(ml-min)/(max-min)
  if (U<F) {min+(U*(max-min)*(ml-min))^.5}
  else {max-((1-U)*(max-min)*(max-ml))^.5}
}


qgen<-function(U)
  ifelse(U<.1, 5000/.1*U,
         ifelse(U<.5, 5000/.4*(U-.1)+5000,
                ifelse(U<.7, 5000/.2*(U-.5)+10000,
                       ifelse(U<.95,5000/.25*(U-.7)+15000,
                              5000/.05*(U-.95)+20000)
                )
         )
  )

x<-runif(100000) %>% sapply(qgen) #testing the qgen function
hist(x)
```



These tasks are accomplished, we can now call the draw.d.variate.uniform() function. It results in a matrix, so we convert it into a data frame for easier manipulation and round the values to their nearest integers. The code for this process is provided in Figure 6.13.

Figure 6.13

```
df <- data.frame(draw.d.variate.uniform(10000, 4, cormatrix2))
colnames(df) <- c("demC", "demF", "demS", "demV")
df$demC <- qnorm(df$demC, cmodel$estimate[1], cmodel$estimate[2])
df$demF <- qweibull(df$demF, fmodel$estimate[1], fmodel$estimate[2])
df$demS <- sapply(df$demS, qgen)
df$demV <- qtri(df$demV, 2500, 10000, 25000)
```

Checking the Input Model

To determine whether our data is correlated correctly, we calculate the Spearman correlations. We also generate scatter plots of each input against each other input to try an eyeball test. Finally, we plot each marginal density to make sure the data follows their original distributions as well.

The results are reproduced in Figures 6.14-6.16.

Figure 6.14

Figure 6.15

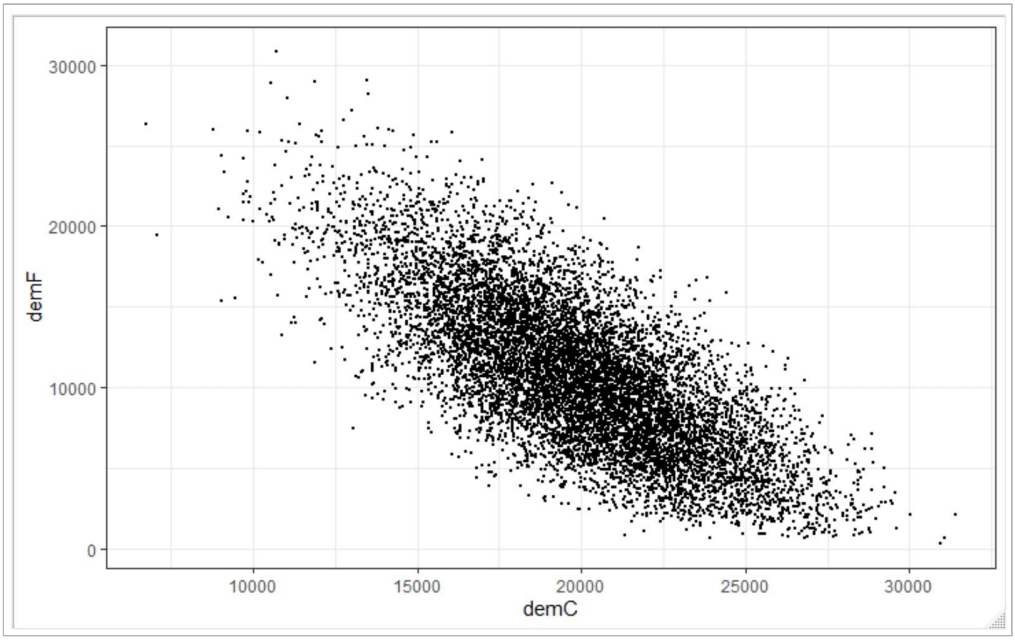


Figure 6.16

The remaining code is exactly the same as it was in the uncorrelated case: we create an output vector of profits in order to study the resulting distribution.

Upon running the correlated simulation, I find a mean profit of $582,000 with a standard deviation of $101,000. Running the uncorrelated version, I find a mean profit of $575,000 with a standard deviation of $111,000. Although these results are quite similar, there is no guarantee this will be the case.

Individual Exercise:

Run the correlated and uncorrelated simulations again, only set the order quantity to 20,000 for each product. Do you still observe a similar result between the two? If there's a difference, how do you account for it?

Lecture 6 References

Camm, J. D., Cochran, J. J., Fry, M. J., Ohlmann, J. W., Anderson, D. R., Sweeney, D. J., & Williams, T. A. (2015). *Essentials of business analytics* (1st edition). Cengage Learning.

Lecture 6 Summary Questions

1. What are the differences between Pearson product moment correlation and Spearman rank correlation?
2. Why use Spearman rank correlation instead of Pearson product moment correlation?
3. Please comment on the following statement: *"It is always a good idea to use the probability distribution with the best fit statistic to model an uncertain variable of interest."*
4. What is the purpose of validating a correlation matrix?
5. Please comment on the following statement: *"It is recommended to take correlations into account when building a simulation model if there is any evidence of correlation among uncertain inputs."*