**Assumptions and Data Sources for Deterministic Variables**

|  |  |  |
| --- | --- | --- |
| Variable | Assumption | Data Source/Rationale |
| Plant capacity | 1,000MW | Equals the average Net Summer Capacity of all US Nuclear Plants currently in operation. |
| Plant lifespan | 60 years | According to the World Nuclear Association, generation III advanced reactors typically have a 60 year operating life. |
| Variable Operating and Maintenance Costs | $2.14/MWh | *Updated Capital Cost Estimates for Utility Scale Electricity Generating Plants*. US Energy Information Administration. 2013. |
| Fuel Costs (incl. uranium, processing, waste disposal) | $6.17/MWh (avg. fuel costs for all US plants, 2002-15) | *Nuclear Costs in Context*. Nuclear Energy Institute. 2016. |
| Fixed Operating and Maintenance Costs | $93.28/kW/yr = $93,280,000/yr for 1,000MW plant | *Updated Capital Cost Estimates for Utility Scale Electricity Generating Plants*. US Energy Information Administration. 2013. |
| Decommissioning costs as % of total overnight costs | 12% | Decommissioning costs are about 9-15% of the initial capital cost of a nuclear power plant. World Nuclear Association. *The Economics of Nuclear Power.* July 2016. |
| Interest rate during construction | 7% | Assumes 100% debt financing and risk premium of 4% |
| Discount rate | 7% | Nuclear projects are typically evaluated at 5% and 10% discount rates. We assume 7% as a mid point. |

**Electricity Prices**

We used U.S. wholesale prices. We originally considered using export prices as a proxy, but that dataset showed a sharp break at the time period that our dataset begins.



**Overnight Construction Costs**

We used construction using all world data. The U.S. has not completed a nuclear power plant since 1980, and it is likely that both past U.S. data and recent world data capture price influences in current U.S. Construction. We considered truncating the data before and after 1980, but either approach omitted too much relevant data.

**Construction Time:**

Looking at the data, we saw no reason to truncate the dataset for time. There is a spike in variance in the early 70s, as with construction costs, but overall the average remains around 6-7 years, and there are spikes occurring into the 2000s.



**Capacity Factor**

We used U.S. data from all plants from 2008 to 2015 to create our distribution.

Appendix Addendum: Distribution Fitting and Monte Carlo Simulation R Code

November 2, 2016

# Distribution Fitting

If you plan on running the below code code you will need to install the necessary packages and any dependencies. If you use Rstudio as the IDE for R, it will automatically install dependencies when the below code is run. Overall commentary is expressed in plain text, code chunks are formatted as greyed out text boxes. Output from code chunks follow after each code chunk.

library(fitdistrplus) #functions for fitting distributions by MoM, MLE, etc.  
library(dplyr) #functions for efficient data wrangling, etc.   
library(actuar) #additional density functions that can be passed to fitting algorithms from fitdistrplus package

#Create construction time metric  
  
construc <- mutate (construc, constr\_time = comm\_op\_year - constr\_year)  
  
elect.prices <- subset(elect.prices, elect.prices$PJM.Wholesale...MWh!="NA") #remove NA rows  
  
elect.price <- data.frame(elect.prices$PJM.Wholesale...MWh) #keep only Wholsale price column  
names(elect.price) <- c("PJM Wholesale") #rename column

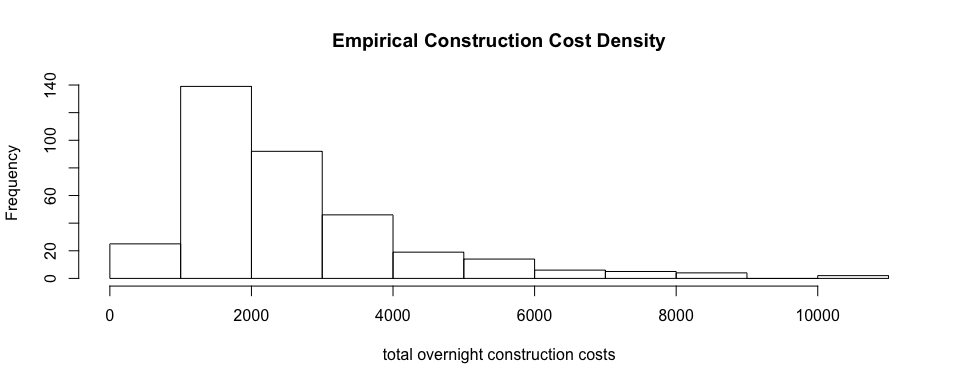
The below code runs through numerous continuous distributions for which maximum likelihood or method of moments (moment matching) could be used to estimate shape and scale parameters. In addition to continuous distributions generally well known, various heavy tailed distributions common in actuarial science, risk theory, etc. are fit. If a well-known distribution is not listed (e.g. beta) it's because it's shape doesn't make sense for the data or there were computational issues that were beyond the scope intended.

The Akaike (AIC) and Bayesian (BIC) information criterion were used as measures of goodness of fit to choose between the distributions so as to not rely solely on visual inspection of diagnostic and fit plots. The final distribution chosen was that which minimized both the AIC and BIC values. Fit measures were used to choose final distributions used because of a lack of theory to guide distributional assumptions despite the risk of over-fitting.

### 

### Distribution Fitting for Overnight Construction Costs

hist <- hist(construc$occ, xlab = "total overnight construction costs", main = "Empirical Construction Cost Density")



#########################################################################  
#########################################################################  
#fit distributions for overnight construction costs ON NON-TRUNCATED DATA  
#########################################################################  
#########################################################################  
  
fitW <- fitdist(construc$occ, "weibull", method = "mle") #weibull  
fitg <- fitdist(construc$occ, "gamma", method = "mme") #gamma ; had to fit using moment-matching  
fitln <- fitdist(construc$occ, "lnorm", method = "mle") #log-normal  
fitn <- fitdist(construc$occ, "norm", method = "mle") #normal  
fitlo <- fitdist(construc$occ, "logis", method = "mle") #logistic  
fitexp <- fitdist(construc$occ, "exp", method = "mme") #exponential  
fitcauchy <- fitdist(construc$occ, "cauchy", method = "mle") #cauchy  
  
fitf <- fitdist(construc$occ, "f", start =c(.0001,.0005), method = "mle") #F-Distribution  
fitt <- fitdist(construc$occ, "t", start = c(.001,.005), method = "mle") #t-distribution (1)  
  
#The below fits using density functions from the actuaral functions package  
#Requires passing the shape and scale parameters as starting values are they are outlined in actuar package documentation as a list  
  
fp <- fitdist(construc$occ, "pareto", start = list(shape = 100, scale = 500)) #pareto  
fitplo <- fitdist(construc$occ, "paralogis", start = list(shape = 1, scale = 500), method = "mle") #para-logistic  
fitllo <- fitdist(construc$occ, "llogis", start = list(shape = 1, scale = 500), method = "mle") #log-logistic  
fitlg <- fitdist(construc$occ, "lgamma", start = list(shapelog = 1, ratelog = 500), method = "mle") #log-gamma  
fIW <- fitdist(construc$occ, "invweibull", start=list(shape = 1, scale = 500), method = "mle") #inverse-weibull  
fItrg <- fitdist(construc$occ, "invtrgamma", start=list(shape1 = 100, shape2 = 100, scale = 500), method = "mle") #inverse-transformed gamma (1)  
fIp <- fitdist(construc$occ, "invpareto", start=list(shape = 200, scale = 500), method = "mle") #inverse-pareto  
fitIplo <- fitdist(construc$occ, "invparalogis", start = list(shape = 1, scale = 500), method = "mle") #inverse para-logistic  
fitIg <- fitdist(construc$occ, "invgamma", start = list(shape = 1, rate = 500), method = "mle") #inverse-gamma  
fitIexp <- fitdist(construc$occ, "invexp", start = list(scale = 500), method = "mle") #inverse-exponential  
  
fIburr <- fitdist(construc$occ, "invburr", start=list(shape1 = 100, shape2 = 100, scale = 500), method = "mle") #inverse-burr  
fGpar <- fitdist(construc$occ, "genpareto", start=list(shape1 = 100, shape2 = 100, scale = 500), method = "mle") #generalized-pareto  
  
  
#Use summary function to return IC statistics and parameter estimates  
  
summary(fitW) #weibull

## Fitting of the distribution ' weibull ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 1.794656 0.06761099  
## scale 2989.761961 94.29342221  
## Loglikelihood: -3039.947 AIC: 6083.894 BIC: 6091.621   
## Correlation matrix:  
## shape scale  
## shape 1.0000000 0.3369458  
## scale 0.3369458 1.0000000

summary(fitg) #gamma

## Fitting of the distribution ' gamma ' by matching moments   
## Parameters :   
## estimate  
## shape 2.713304879  
## rate 0.001028119  
## Loglikelihood: -3022.129 AIC: 6048.257 BIC: 6055.985

summary(fitln) #log-normal

## Fitting of the distribution ' lnorm ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## meanlog 7.7282084 0.02865636  
## sdlog 0.5376411 0.02026279  
## Loglikelihood: -3001.357 AIC: 6006.714 BIC: 6014.442   
## Correlation matrix:  
## meanlog sdlog  
## meanlog 1 0  
## sdlog 0 1

summary(fitn) #normal

## Fitting of the distribution ' norm ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## mean 2639.097 85.40263  
## sd 1602.160 60.37627  
## Loglikelihood: -3096.912 AIC: 6197.825 BIC: 6205.552   
## Correlation matrix:  
## mean sd  
## mean 1 0  
## sd 0 1

summary(fitlo) #logistic

## Fitting of the distribution ' logis ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## location 2395.5543 72.42728  
## scale 796.8174 36.33959  
## Loglikelihood: -3067.702 AIC: 6139.403 BIC: 6147.13   
## Correlation matrix:  
## location scale  
## location 1.0000000 0.1116945  
## scale 0.1116945 1.0000000

summary(fitexp) #exponential

## Fitting of the distribution ' exp ' by matching moments   
## Parameters :   
## estimate  
## rate 0.0003789175  
## Loglikelihood: -3125.124 AIC: 6252.247 BIC: 6256.111

summary(fitcauchy) #cauchy

## Fitting of the distribution ' cauchy ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## location 2018.591 53.07718  
## scale 633.460 46.87442  
## Loglikelihood: -3068.101 AIC: 6140.201 BIC: 6147.929   
## Correlation matrix:  
## location scale  
## location 1.000000 0.276617  
## scale 0.276617 1.000000

summary(fitf) #F

## Fitting of the distribution ' f ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## 1 4.106172e+05 10.85370784  
## 2 2.362895e-01 0.01326496  
## Loglikelihood: -3862.205 AIC: 7728.409 BIC: 7736.136   
## Correlation matrix:  
## [,1] [,2]  
## [1,] 1.000000000 -0.001485994  
## [2,] -0.001485994 1.000000000

summary(fitt) #t

## Fitting of the distribution ' t ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## 1 2.519285 0.1671066  
## 2 1641.574587 42.7567302  
## Loglikelihood: -3027.631 AIC: 6059.262 BIC: 6066.989   
## Correlation matrix:  
## [,1] [,2]  
## [1,] 1.0000000 -0.1624978  
## [2,] -0.1624978 1.0000000

summary(fp) #pareto

## Fitting of the distribution ' pareto ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 170796.4 NA  
## scale 450571159.5 NA  
## Loglikelihood: -3125.124 AIC: 6254.248 BIC: 6261.976   
## Correlation matrix:  
## [1] NA

summary(fitplo) #para-logistic

## Fitting of the distribution ' paralogis ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 2.428692 0.08297592  
## scale 3647.997499 113.47204890  
## Loglikelihood: -3014.215 AIC: 6032.429 BIC: 6040.157   
## Correlation matrix:  
## shape scale  
## shape 1.0000000 0.2679846  
## scale 0.2679846 1.0000000

summary(fitllo) #log-logistic

## Fitting of the distribution ' llogis ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 3.289693 0.1460799  
## scale 2234.458000 63.0632810  
## Loglikelihood: -3002.7 AIC: 6009.4 BIC: 6017.128   
## Correlation matrix:  
## shape scale  
## shape 1.00000000 -0.03185524  
## scale -0.03185524 1.00000000

summary(fitlg) #log-gamma

## Fitting of the distribution ' lgamma ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shapelog 204.91932 15.433818  
## ratelog 26.51561 1.999503  
## Loglikelihood: -3000.213 AIC: 6004.427 BIC: 6012.154   
## Correlation matrix:  
## shapelog ratelog  
## shapelog 1.0000000 0.9987803  
## ratelog 0.9987803 1.0000000

summary(fIW) #inverse-weibull

## Fitting of the distribution ' invweibull ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 1.940078 0.07422245  
## scale 1744.719825 50.79626301  
## Loglikelihood: -3019.286 AIC: 6042.572 BIC: 6050.299   
## Correlation matrix:  
## shape scale  
## shape 1.0000000 -0.3311958  
## scale -0.3311958 1.0000000

summary(fItrg) #inverse-transformed gamma

## Fitting of the distribution ' invtrgamma ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape1 0.2273161 0.03005205  
## shape2 3.7866647 0.44661801  
## scale 670.5321026 39.29583278  
## Loglikelihood: -3132.476 AIC: 6270.952 BIC: 6282.542   
## Correlation matrix:  
## shape1 shape2 scale  
## shape1 1.0000000 -0.84824063 0.27825393  
## shape2 -0.8482406 1.00000000 0.09140288  
## scale 0.2782539 0.09140288 1.00000000

summary(fIp) #inverse-pareto

## Fitting of the distribution ' invpareto ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 442.129931 284.108663  
## scale 4.459813 2.870011  
## Loglikelihood: -3121.717 AIC: 6247.433 BIC: 6255.161   
## Correlation matrix:  
## shape scale  
## shape 1.000000 -0.996554  
## scale -0.996554 1.000000

summary(fitIplo) #inverse-para-logistic

## Fitting of the distribution ' invparalogis ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 2.564832 0.09116499  
## scale 1381.867454 39.87028934  
## Loglikelihood: -3001.864 AIC: 6007.729 BIC: 6015.456   
## Correlation matrix:  
## shape scale  
## shape 1.000000 -0.269571  
## scale -0.269571 1.000000

summary(fitIg) #inverse-gamma

## Fitting of the distribution ' invgamma ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 0.082773196 0.004553057  
## rate 0.006118906 0.001082440  
## Loglikelihood: -3688.279 AIC: 7380.557 BIC: 7388.284   
## Correlation matrix:  
## shape rate  
## shape 1.0000000 -0.2658596  
## rate -0.2658596 1.0000000

summary(fitIexp) #inverse-exponential

## Fitting of the distribution ' invexp ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## scale 1974.991 105.253  
## Loglikelihood: -3121.438 AIC: 6244.876 BIC: 6248.739

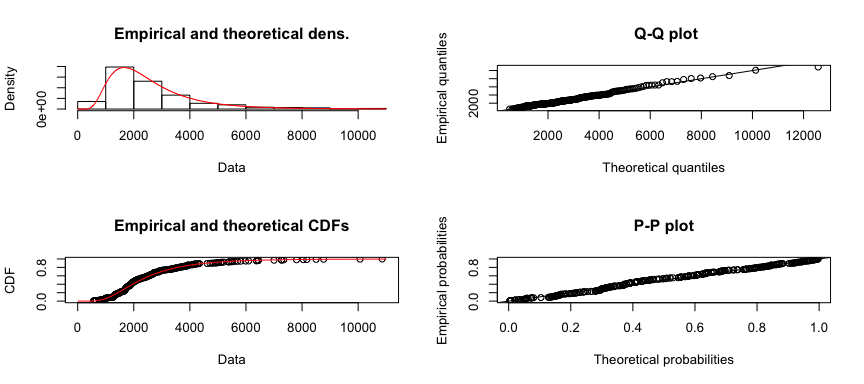
summary(fIburr) #inverse-burr

## Fitting of the distribution ' invburr ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape1 39.460482 NaN  
## shape2 1.981286 0.07164032  
## scale 276.513221 NaN  
## Loglikelihood: -3017.073 AIC: 6040.145 BIC: 6051.736   
## Correlation matrix:  
## shape1 shape2 scale  
## shape1 1 NaN NaN  
## shape2 NaN 1 NaN  
## scale NaN NaN 1

summary(fGpar) #generalized-pareto

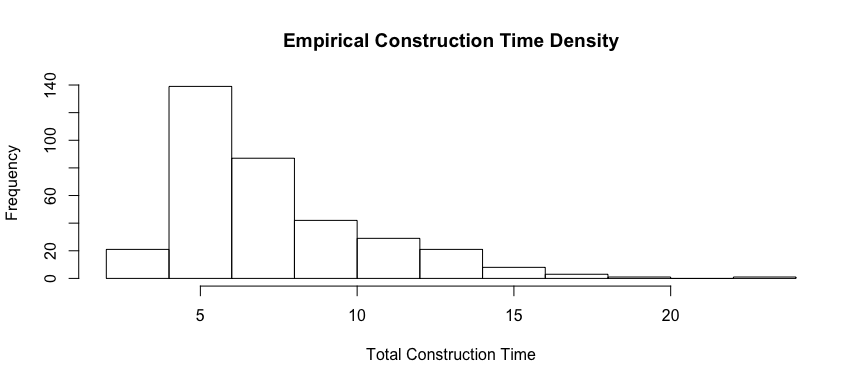
## Fitting of the distribution ' genpareto ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape1 5.266885 0.8612065  
## shape2 12.846111 4.7666303  
## scale 878.153230 473.1225648  
## Loglikelihood: -2999.897 AIC: 6005.793 BIC: 6017.384   
## Correlation matrix:  
## shape1 shape2 scale  
## shape1 1.0000000 -0.7903252 0.8940331  
## shape2 -0.7903252 1.0000000 -0.9791653  
## scale 0.8940331 -0.9791653 1.0000000

#Use plot functions to create fit and diagnostic plots for visual inspection of fit  
plot(fitlg) #best fit(s) per Information Criterion



### Distribution Fitting for Construction Time

hist <- hist(construc$constr\_time, xlab = "Total Construction Time", main = "Empirical Construction Time Density")



########################################  
########################################  
#fit distributions for construction time  
########################################  
########################################  
  
fitW <- fitdist(construc$constr\_time, "weibull", method = "mle") #weibull  
fitg <- fitdist(construc$constr\_time, "gamma", method = "mme") #gamma ; had to fit using moment-matching  
fitln <- fitdist(construc$constr\_time, "lnorm", method = "mle") #log-normal  
fitn <- fitdist(construc$constr\_time, "norm", method = "mle") #normal  
fitlo <- fitdist(construc$constr\_time, "logis", method = "mle") #logistic  
fitexp <- fitdist(construc$constr\_time, "exp", method = "mme") #exponential  
fitcauchy <- fitdist(construc$constr\_time, "cauchy", method = "mle") #cauchy  
  
fitf <- fitdist(construc$constr\_time, "f", start = c(.000001,.0000005), method = "mle") #F-Distribution  
fitt <- fitdist(construc$constr\_time, "t", start = c(1,5), method = "mle") #t-distribution (1)  
  
#The below fits using density functions from the actuaral functions package  
#Requires passing the shape and scale parameters as starting values are they are outlined in actuar package documentation as a list  
  
fp <- fitdist(construc$constr\_time, "pareto", start = list(shape = 100, scale = 500)) #pareto  
fitplo <- fitdist(construc$constr\_time, "paralogis", start = list(shape = 1, scale = 500), method = "mle") #para-logistic  
fitllo <- fitdist(construc$constr\_time, "llogis", start = list(shape = 1, scale = 500), method = "mle") #log-logistic  
fitlg <- fitdist(construc$constr\_time, "lgamma", start = list(shapelog = 1, ratelog = 500), method = "mle") #log-gamma  
fIW <- fitdist(construc$constr\_time, "invweibull", start=list(shape = 1, scale = 500), method = "mle") #inverse-weibull  
fItrg <- fitdist(construc$constr\_time, "invtrgamma", start=list(shape1 = 1, shape2 = 1, scale = 500), method = "mle") #inverse-transformed gamma (1)  
fIp <- fitdist(construc$constr\_time, "invpareto", start=list(shape = 200, scale = 500), method = "mle") #inverse-pareto  
fitIplo <- fitdist(construc$constr\_time, "invparalogis", start = list(shape = 1, scale = 500), method = "mle") #inverse para-logistic  
fitIg <- fitdist(construc$constr\_time, "invgamma", start = list(shape = 1, rate = 500), method = "mle") #inverse-gamma  
fitIexp <- fitdist(construc$constr\_time, "invexp", start = list(scale = 500), method = "mle") #inverse-exponential  
  
fIburr <- fitdist(construc$constr\_time, "invburr", start=list(shape1 = 100, shape2 = 100, scale = 500), method = "mle") #inverse-burr  
fGpar <- fitdist(construc$constr\_time, "genpareto", start=list(shape1 = 100, shape2 = 100, scale = 500), method = "mle") #generalized-pareto  
  
  
#Use summary function to return IC statistics and parameter estimates  
  
summary(fitW) #weibull

## Fitting of the distribution ' weibull ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 2.496435 0.09349168  
## scale 8.211467 0.18642397  
## Loglikelihood: -873.5489 AIC: 1751.098 BIC: 1758.825   
## Correlation matrix:  
## shape scale  
## shape 1.0000000 0.3398472  
## scale 0.3398472 1.0000000

summary(fitg) #gamma

## Fitting of the distribution ' gamma ' by matching moments   
## Parameters :   
## estimate  
## shape 5.7516681  
## rate 0.7906082  
## Loglikelihood: -841.9862 AIC: 1687.972 BIC: 1695.7

summary(fitln) #log-normal

## Fitting of the distribution ' lnorm ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## meanlog 1.9110167 0.01979381  
## sdlog 0.3713649 0.01399588  
## Loglikelihood: -823.4635 AIC: 1650.927 BIC: 1658.654   
## Correlation matrix:  
## meanlog sdlog  
## meanlog 1 0  
## sdlog 0 1

summary(fitn) #normal

## Fitting of the distribution ' norm ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## mean 7.274991 0.1616830  
## sd 3.033441 0.1143271  
## Loglikelihood: -890.0799 AIC: 1784.16 BIC: 1791.887   
## Correlation matrix:  
## mean sd  
## mean 1 0  
## sd 0 1

summary(fitlo) #logistic

## Fitting of the distribution ' logis ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## location 6.841395 0.14800191  
## scale 1.606980 0.07266432  
## Loglikelihood: -876.1523 AIC: 1756.305 BIC: 1764.032   
## Correlation matrix:  
## location scale  
## location 1.0000000 0.1193447  
## scale 0.1193447 1.0000000

summary(fitexp) #exponential

## Fitting of the distribution ' exp ' by matching moments   
## Parameters :   
## estimate  
## rate 0.1374572  
## Loglikelihood: -1050.524 AIC: 2103.048 BIC: 2106.911

summary(fitcauchy) #cauchy

## Fitting of the distribution ' cauchy ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## location 6.035302 0.11180355  
## scale 1.340303 0.09832073  
## Loglikelihood: -893.9077 AIC: 1791.815 BIC: 1799.543   
## Correlation matrix:  
## location scale  
## location 1.000000 0.252587  
## scale 0.252587 1.000000

summary(fitf) #F

## Fitting of the distribution ' f ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## 1 5.327228e+05 2.971541e+03  
## 2 1.162617e+00 7.351667e-02  
## Loglikelihood: -1357.526 AIC: 2719.051 BIC: 2726.778   
## Correlation matrix:  
## [,1] [,2]  
## [1,] 1.00000000 -0.06201763  
## [2,] -0.06201763 1.00000000

summary(fitt) #t

## Fitting of the distribution ' t ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## 1 5.353522 0.4561821  
## 2 6.222929 0.1131867  
## Loglikelihood: -815.1894 AIC: 1634.379 BIC: 1642.106   
## Correlation matrix:  
## [,1] [,2]  
## [1,] 1.00000000 0.08421494  
## [2,] 0.08421494 1.00000000

summary(fp) #pareto

## Fitting of the distribution ' pareto ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 21450984 0.00  
## scale 156045672 20547.81  
## Loglikelihood: -1050.524 AIC: 2105.048 BIC: 2112.775   
## Correlation matrix:  
## shape scale  
## shape 1 Inf  
## scale Inf 1

summary(fitplo) #para-logistic

## Fitting of the distribution ' paralogis ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 3.170382 0.1089772  
## scale 10.676402 0.2369770  
## Loglikelihood: -850.5858 AIC: 1705.172 BIC: 1712.899   
## Correlation matrix:  
## shape scale  
## shape 1.0000000 0.1694853  
## scale 0.1694853 1.0000000

summary(fitllo) #log-logistic

## Fitting of the distribution ' llogis ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 4.661796 0.2055653  
## scale 6.583817 0.1322348  
## Loglikelihood: -829.2014 AIC: 1662.403 BIC: 1670.13   
## Correlation matrix:  
## shape scale  
## shape 1.00000000 -0.06366217  
## scale -0.06366217 1.00000000

summary(fitlg) #log-gamma

## Fitting of the distribution ' lgamma ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shapelog 27.52668 2.062454  
## ratelog 14.39567 1.088475  
## Loglikelihood: -813.0087 AIC: 1630.017 BIC: 1637.745   
## Correlation matrix:  
## shapelog ratelog  
## shapelog 1.0000000 0.9909328  
## ratelog 0.9909328 1.0000000

summary(fIW) #inverse-weibull

## Fitting of the distribution ' invweibull ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 3.265239 0.1363226  
## scale 5.677805 0.0978014  
## Loglikelihood: -809.2382 AIC: 1622.476 BIC: 1630.204   
## Correlation matrix:  
## shape scale  
## shape 1.000000 -0.320918  
## scale -0.320918 1.000000

summary(fItrg) #inverse-transformed gamma

## Fitting of the distribution ' invtrgamma ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape1 17.6757605 NaN  
## shape2 0.6614839 NaN  
## scale 497.7568858 NaN  
## Loglikelihood: -816.639 AIC: 1639.278 BIC: 1650.869   
## Correlation matrix:  
## shape1 shape2 scale  
## shape1 1 NaN NaN  
## shape2 NaN 1 NaN  
## scale NaN NaN 1

summary(fIp) #inverse-pareto

## Fitting of the distribution ' invpareto ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 183.20688636 69.0892967  
## scale 0.03439583 0.0129988  
## Loglikelihood: -1048.069 AIC: 2100.138 BIC: 2107.865   
## Correlation matrix:  
## shape scale  
## shape 1.0000000 -0.9899613  
## scale -0.9899613 1.0000000

summary(fitIplo) #inverse-para-logistic

## Fitting of the distribution ' invparalogis ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 3.72898 0.14228331  
## scale 4.19754 0.07480489  
## Loglikelihood: -812.6128 AIC: 1629.226 BIC: 1636.953   
## Correlation matrix:  
## shape scale  
## shape 1.0000000 -0.1282249  
## scale -0.1282249 1.0000000

summary(fitIg) #inverse-gamma

## Fitting of the distribution ' invgamma ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 7.97336395 0.536594025  
## rate 0.01978199 0.001363724  
## Loglikelihood: -814.0949 AIC: 1632.19 BIC: 1639.917   
## Correlation matrix:  
## shape rate  
## shape 1.000000 -0.962369  
## rate -0.962369 1.000000

summary(fitIexp) #inverse-exponential

## Fitting of the distribution ' invexp ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## scale 6.340774 0.3379644  
## Loglikelihood: -1047.211 AIC: 2096.423 BIC: 2100.287

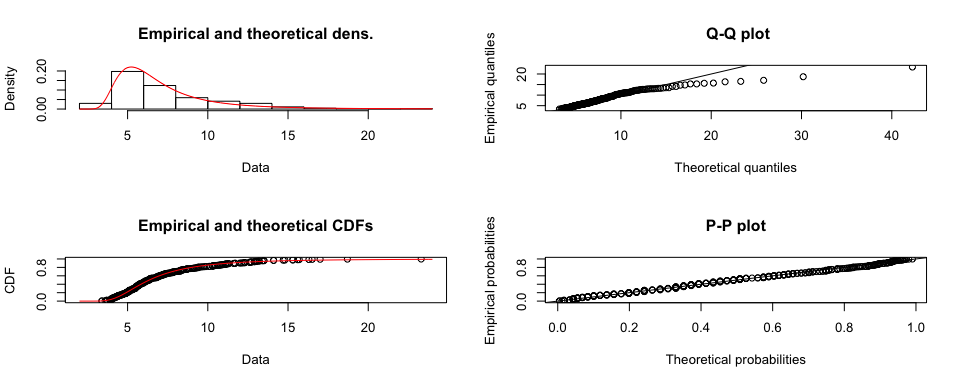
summary(fIburr) #inverse-burr

## Fitting of the distribution ' invburr ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape1 0.1252188 0.03117424  
## shape2 1.7826123 0.43650523  
## scale 573.8190749 NaN  
## Loglikelihood: -1549.688 AIC: 3105.376 BIC: 3116.967   
## Correlation matrix:  
## shape1 shape2 scale  
## shape1 1.0000000 -0.9811243 NaN  
## shape2 -0.9811243 1.0000000 NaN  
## scale NaN NaN 1

summary(fGpar) #generalized-pareto

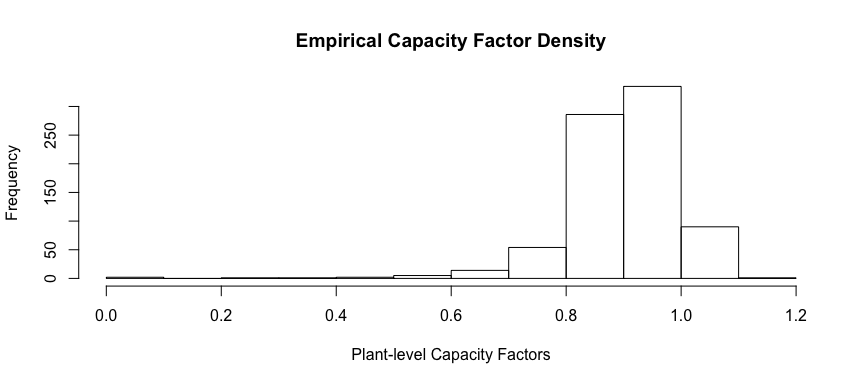
## Fitting of the distribution ' genpareto ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape1 373.358784 NaN  
## shape2 7.129142 0.5187013  
## scale 379.700015 NaN  
## Loglikelihood: -838.0787 AIC: 1682.157 BIC: 1693.748   
## Correlation matrix:  
## shape1 shape2 scale  
## shape1 1 NaN NaN  
## shape2 NaN 1 NaN  
## scale NaN NaN 1

plot(fIW) #best fit per Information Criterion



### Distribution Fitting for Capacity Factor

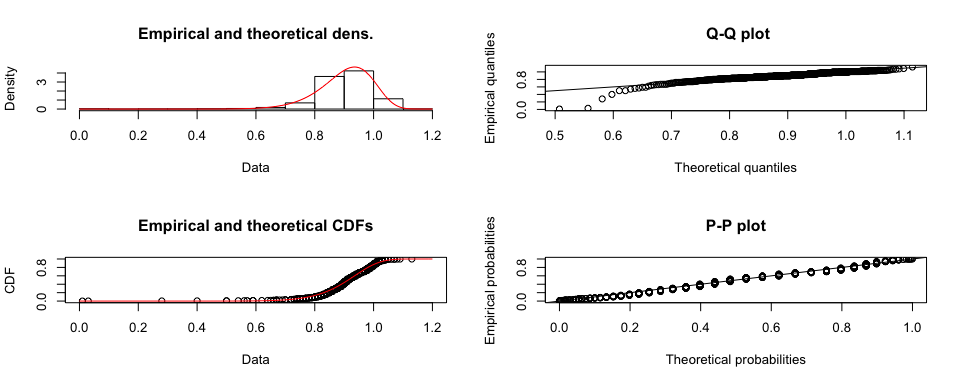
hist <- hist(capacity.Factor$percentage, xlab = "Plant-level Capacity Factors", main = "Empirical Capacity Factor Density")



########################################  
########################################  
#fit distribution for Capacity Factor  
########################################  
########################################  
  
  
#######ONLY SHOWS BEST FIT DISTRIBUTION##########  
  
fitW <- fitdist(capacity.Factor$percentage, "weibull", method = "mle") #weibull  
summary(fitW)

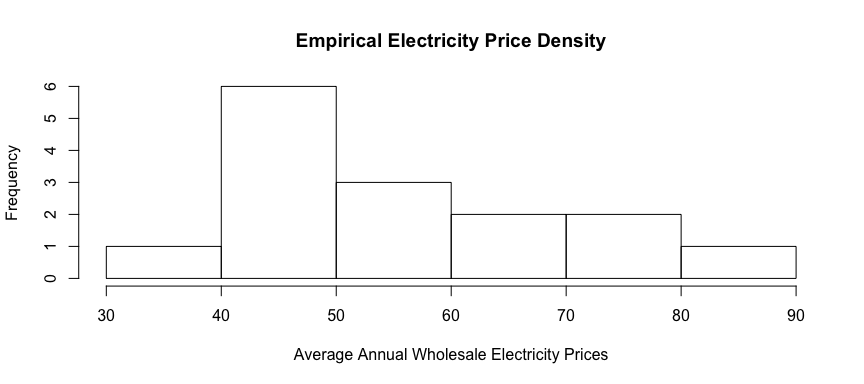
## Fitting of the distribution ' weibull ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 11.9019211 0.340368656  
## scale 0.9422547 0.002931445  
## Loglikelihood: 752.5012 AIC: -1501.002 BIC: -1491.656   
## Correlation matrix:  
## shape scale  
## shape 1.0000000 0.2792129  
## scale 0.2792129 1.0000000

plot(fitW) #best fit per Information Criterion



### Distribution Fitting for Electricity Prices

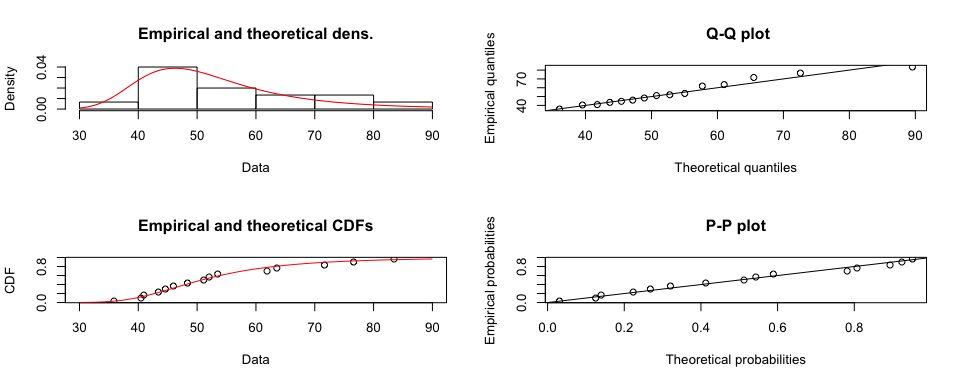
hist <- hist(elect.price$`PJM Wholesale`, xlab = "Average Annual Wholesale Electricity Prices", main = "Empirical Electricity Price Density")



########################################  
########################################  
#fit distribution for Electricity Prices  
########################################  
########################################  
  
  
#######ONLY SHOWS BEST FIT DISTRIBUTION##########  
  
fitIplo <- fitdist(elect.price$`PJM Wholesale`, "invparalogis", start = list(shape = 1, scale = 500), method = "mle") #inverse para-logistic  
summary(fitIplo)

## Fitting of the distribution ' invparalogis ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## shape 5.423684 1.032145  
## scale 35.153542 1.963789  
## Loglikelihood: -59.14384 AIC: 122.2877 BIC: 123.7038   
## Correlation matrix:  
## shape scale  
## shape 1.0000000 0.1101445  
## scale 0.1101445 1.0000000

plot(fitIplo)



### Understanding Random Covariate Draws from Chosen Distributions

#Draw 10k covariates from fitted distributions to get a sense of possible values that can be drawn for simulation -- output and review in Excel  
  
time.draw <- rinvweibull(10000, fIW$estimate[1], rate = 1, scale = fIW$estimate[2]) #draws 1 each simulation  
  
occ.draw <- rlgamma(10000, shapelog = fitlg$estimate[1], ratelog = fitlg$estimate[2]) #draws 1 each simulation from truncated dist  
  
Capac.draw <- rweibull(10000, shape = fitW$estimate[1], scale = fitW$estimate[2]) #draws 60 each simulation   
  
Price.draw <- rinvparalogis(10000, shape = fitIplo$estimate[1], rate = 1, scale = fitIplo$estimate[2]) #draws 60 each simulation  
  
# Load workbook (create if not existing)  
wb <- loadWorkbook("/Users/Unholi/Documents/JohnsHopkins\_MSApplied Economics/JHU Real Risk\_FA16/R/Project/Random Draws.xlsx", create = TRUE)  
  
time.df <- data.frame(time.draw) #output data has to be data frame  
occ.df <- as.data.frame(occ.draw)  
capac.df <- as.data.frame(Capac.draw)  
price.df <- as.data.frame(Price.draw)  
  
# Create a worksheet called 'time'  
createSheet(wb, name = "time")  
  
# Create a worksheet called 'occ'  
createSheet(wb, name = "occ")  
  
# Create a worksheet called 'capacity'  
createSheet(wb, name = "capacity")  
  
# Create a worksheet called 'price'  
createSheet(wb, name = "price")  
  
writeWorksheet(wb, time.df, sheet="time", startRow = 1, startCol = 1, header = TRUE)   
writeWorksheet(wb, occ.df, sheet = "occ", startRow = 1, startCol = 1)  
writeWorksheet(wb, capac.df, sheet = "capacity", startRow = 1, startCol = 1)  
writeWorksheet(wb, price.df, sheet = "price", startRow = 1, startCol = 1)  
  
saveWorkbook(wb)

# Monte Carlo Simulation

The Monte Carlo Simulation facilitates varying of the uncertainties in our specified value function for NPV so that in the end we get a distribution of NPV from which we can make appropriate inferences.

#########################################################################################################################  
#Use Monte Carlo Simulation to bake uncertainties into Net Present Value Calculation for Midterm Risk Assesment Project##  
#########################################################################################################################  
  
NPV.dist <- as.numeric(c(rep(0,10000))) #create empty storage vector to hold final NPV for each trial  
CC.dist <- as.numeric(c(rep(0,10000))) #create empty storage vector to hold final capital cost for each trial  
OP.dist <- as.numeric(c(rep(0,10000))) #create empty storage vector to hold final operating profit for each trial  
DECOM.dist <- as.numeric(c(rep(0,10000))) #create empty storage vector to hold final decommission cost for each trial  
  
r <- .07 #discount rate for each trial  
  
for (i in 1:10000) { #each loop is 1 trial in the simulation  
   
######################  
####CAPITAL COST######  
######################  
  
time.draw <- 25 #truncate inverse weibull distribution so that time draw is less than 25 years  
  
while (time.draw > 24) {  
  
time.draw <- rinvweibull(1, fIW$estimate[1], rate = 1, scale = fIW$estimate[2]) #draws 1 each simulation from truncated dist  
  
}  
  
cost.vec <- as.numeric(c(rep(0,round(time.draw)))) #create empty storage vector to store cost results  
occ.draw.vect <- as.numeric(c(rep(0,round(time.draw)))) #create empty storage vector to store occ draw results  
  
occ.draw <- 11000 #truncate log-gamma distribution so that occ draw is less than 11 billion  
  
while (occ.draw > 10999) {  
   
 occ.draw <- rlgamma(1, shapelog = fitlg$estimate[1], ratelog = fitlg$estimate[2]) #draws 1 each simulation from truncated dist  
   
}  
  
for (t in 1 : round(time.draw)) {  
   
 x <- (1 / round(time.draw)) \* (occ.draw \* 1000000) \* ((1 + r)^(round(time.draw) - t + 1)) # occ + financing costs  
   
 cost.vec[t] <- x #stores results in vector to be summed over to arrive at total Capital Cost for the reactor  
   
 #store draw result for occ for 1 trial; should be the same for each loop or you are drawing more than 1 occ  
 occ.draw.vect[t] <- occ.draw  
   
 }  
  
Capital.Cost <- sum(cost.vec)  
  
CC.dist[i] <- Capital.Cost  
  
############################  
####DECOMMISSIONING COST####  
############################  
  
Decommission.Cost <- (.12 \* (occ.draw \* 1000000)) / (1 + r)^61 #1-time decommission cost  
  
DECOM.dist[i] <- Decommission.Cost  
  
########################  
####OPERATING PROFIT####  
########################  
  
Gen.vec <- as.numeric(c(rep(0,60))) #create empty storeage vector  
capac.draw.vec <- as.numeric(c(rep(0,60))) #create empty storeage vector  
price.draw.vec <- as.numeric(c(rep(0,60))) #create empty storeage vector  
Operat.vec <- as.numeric(c(rep(0,60))) #create empty storeage vector  
  
for (j in 1:60) { #60 years operating life  
   
 #draws 60 each simulation since draw is within operating life for loop nested within simulation for loop  
 Capac.draw <- rweibull(1, shape = fitW$estimate[1], scale = fitW$estimate[2])   
   
 #draws 60 each simulation since draw is within operating life for loop nested within simulation for loop  
 Price.draw <- rinvparalogis(1, shape = fitIplo$estimate[1], rate = 1, scale = fitIplo$estimate[2])   
   
 Gen <- (1000 \* 8760 \* Capac.draw)  
   
 #vector to be summed over to arrive at operating profit over the life of the reactor  
 Operat.vec[j] <- ((Gen \* Price.draw) - (Gen \* 8.31) - 93280000) / (1 + r)^j   
   
 #store price and capacity factor draws as well as generation calculation for audit purposes  
 capac.draw.vec[j] <- Capac.draw  
 price.draw.vec[j] <- Price.draw  
 Gen.vec[j] <- Gen   
   
 }  
  
 Operating.Profit <- sum(Operat.vec)  
   
 OP.dist[i] <- Operating.Profit  
  
##################  
#NET PRESENT VALUE  
##################  
  
 NPV.dist[i] <- Operating.Profit - Capital.Cost - Decommission.Cost  
  
  
} #END Simulation