

Exam 2018-05-16

Maximilian Pfundstein

2019-03-12

Contents

1	Assignment 1	1
1.1	Question 1.1	1
2	Assignment 2	5

1 Assignment 1

1.1 Question 1.1

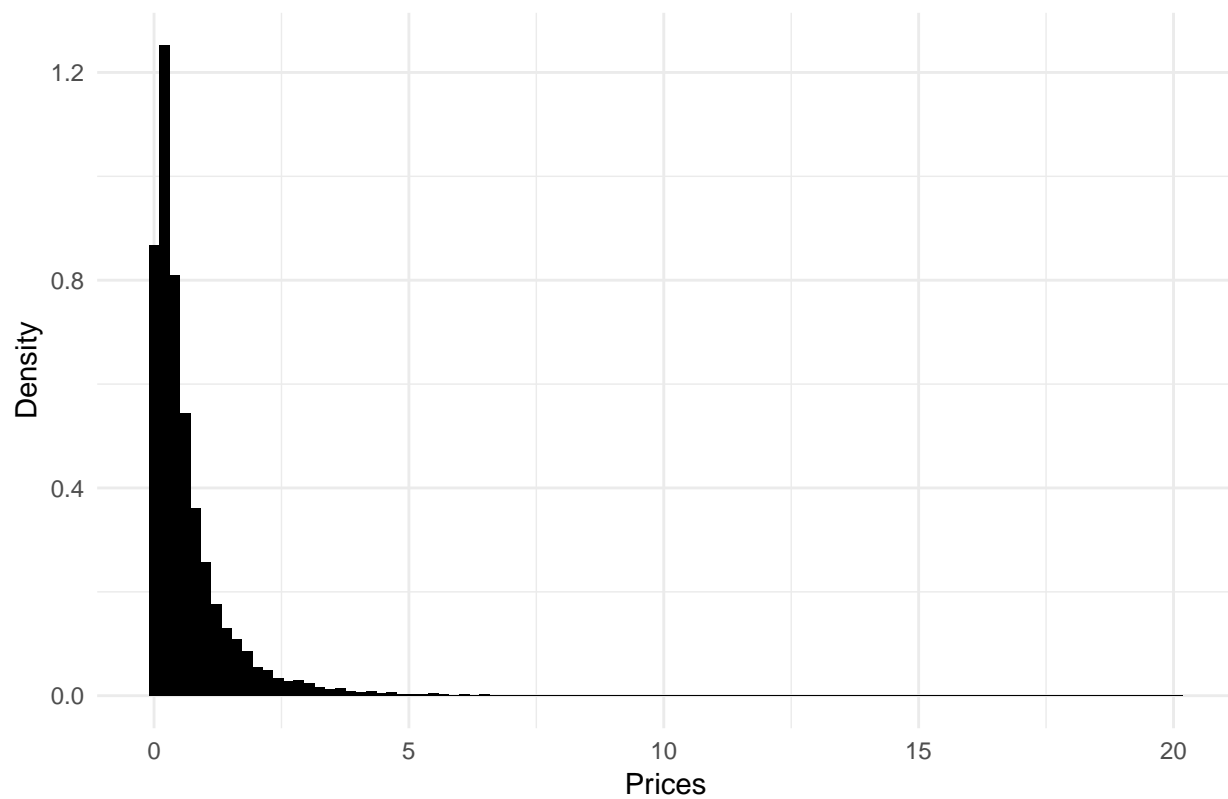
Let's first sample from this distribution and plot the density to get a feeling for it.

Note: Is the `dpareto()` wrong? Should it not first produce values other than zeros when $x > \beta$?

```
samples_true = rpareto(10000, 4, 2)
samples_true_df = data.frame(samples_true)

ggplot(samples_true_df) +
  geom_histogram(aes(x = samples_true, y=..density..),
                 fill = "black", bins = 100) +
  labs(title = "The Generalized Pareto Distribution",
       y = "Density",
       x = "Prices", color = "Legend") +
  theme_minimal()
```

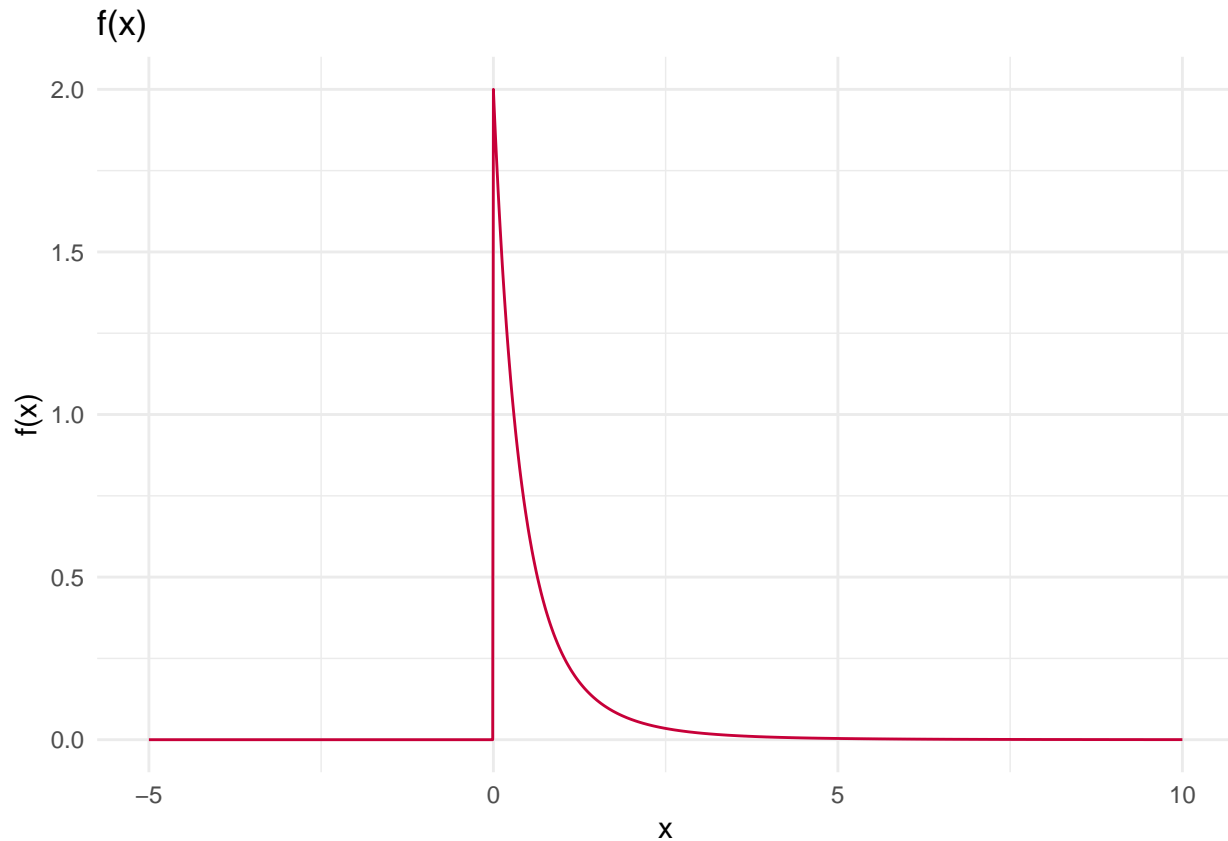
The Generalized Pareto Distribution



```
sequence = seq(from = -5, to = 10, by = 0.01)
pareto_density = sapply(sequence, dpareto, shape = 4, scale = 2)
#pareto_density = sapply(sequence, dpar, alpha = 4, beta = 2)

df = data.frame(sequence, pareto_density)

ggplot(df) +
  geom_line(aes(x = sequence, y = pareto_density), color = "#C70039") +
  labs(title = "f(x)", y = "f(x)", x = "x") +
  theme_minimal()
```



We don't have to integrate over the full domain, as the lower limit is given by β . If $\beta = c$, then x is also given by $x > c$. Thus we have to integrate from β to x .

$$F(x) = \int_{\beta}^x \alpha \beta^{\alpha} \frac{1}{x^{\alpha+1}} = \dots = 1 - \frac{\beta^{\alpha}}{x^{\alpha}}$$

The inverse function Q is obtained by swapping x and y and then solving for y .

$$Q(x) = \frac{\beta}{\sqrt[\alpha]{x-1}}$$

```
# PDF
dpar = function(x, alpha = 3, beta = 1) {
  if (x > beta && alpha > 0 && beta > 0) {
    return(alpha * beta^alpha / (x^alpha + 1))
  }
  return(0)
}

# CDF
ppar = function(x, alpha = 3, beta = 1) {
  if (x <= 1) return(0)
  return(1 - (beta^alpha/x^alpha))
}

# Inverse
qqpar = function(x, alpha = 3, beta = 1) {
  return(beta/((1-x)^(1/alpha)))
}
```

```

}

# Random
rpar = function(n = 1, alpha = 3, beta = 1) {
  quantiles = runif(n = n, min = 0, max = 1)
  rpars = sapply(X = quantiles, FUN = qpar, alpha, beta)
  return(rpars)
}

#sample_rpar_1 = rpar(10000, alpha = 3, beta = 1)
sample_rpar_1 = rgenpareto(10000, shape1 = 3, shape2 = 1)
sample_rpar_2 = rpar(10000, alpha = 4, beta = 2)
sample_rpar_3 = rpar(10000, alpha = 4, beta = 4)
sample_rpar_4 = rpar(10000, alpha = 2, beta = 4)

df = data.frame(sample_rpar_1, sample_rpar_2, sample_rpar_3, sample_rpar_4)

p1 = ggplot(df) +
  geom_histogram(aes(x = sample_rpar_1, y=..density..),
    fill = "#3900c7", bins = 100) +
  ggtitle("alpha = 3, beta = 1") +
  theme_minimal() #+ xlim(0, 2) + ylim(0, 10)

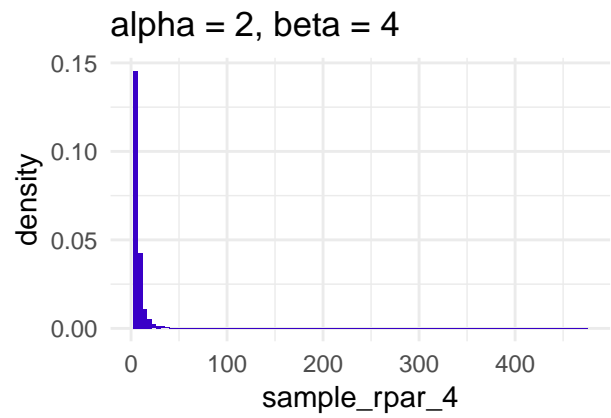
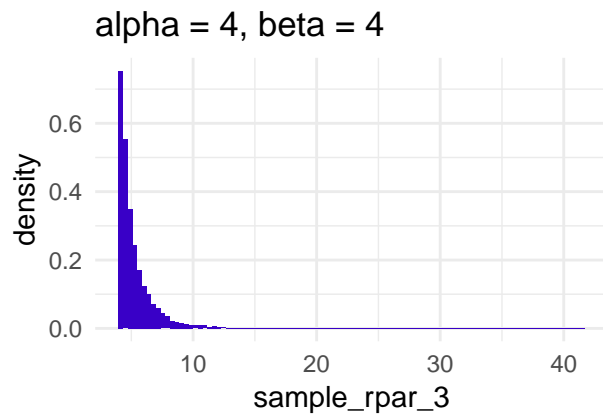
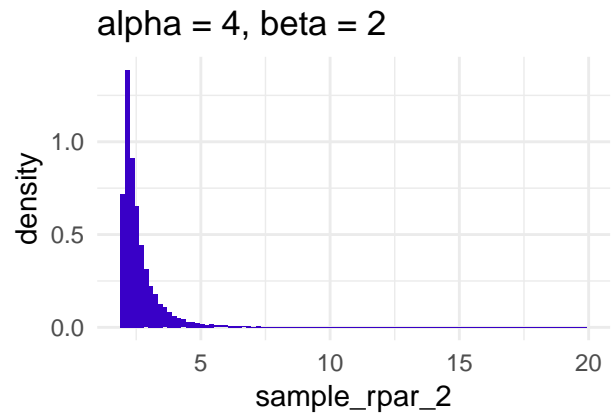
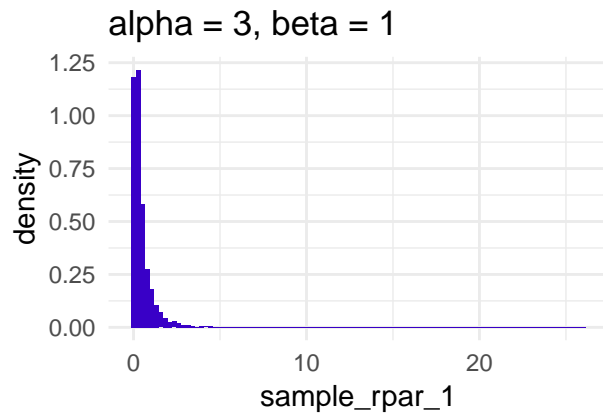
p2 = ggplot(df) +
  geom_histogram(aes(x = sample_rpar_2, y=..density..),
    fill = "#3900c7", bins = 100) +
  ggtitle("alpha = 4, beta = 2") +
  theme_minimal() #+ xlim(0, 2) + ylim(0, 10)

p3 = ggplot(df) +
  geom_histogram(aes(x = sample_rpar_3, y=..density..),
    fill = "#3900c7", bins = 100) +
  ggtitle("alpha = 4, beta = 4") +
  theme_minimal() #+ xlim(0, 2) + ylim(0, 10)

p4 = ggplot(df) +
  geom_histogram(aes(x = sample_rpar_4, y=..density..),
    fill = "#3900c7", bins = 100) +
  ggtitle("alpha = 2, beta = 4") +
  theme_minimal() #+ xlim(0, 2) + ylim(0, 10)

grid.arrange(p1, p2, p3, p4, nrow = 2)

```



2 Assignment 2