Spring 2017 STAT 151A - HW2

Xuanpei Ouyang

February 8, 2017

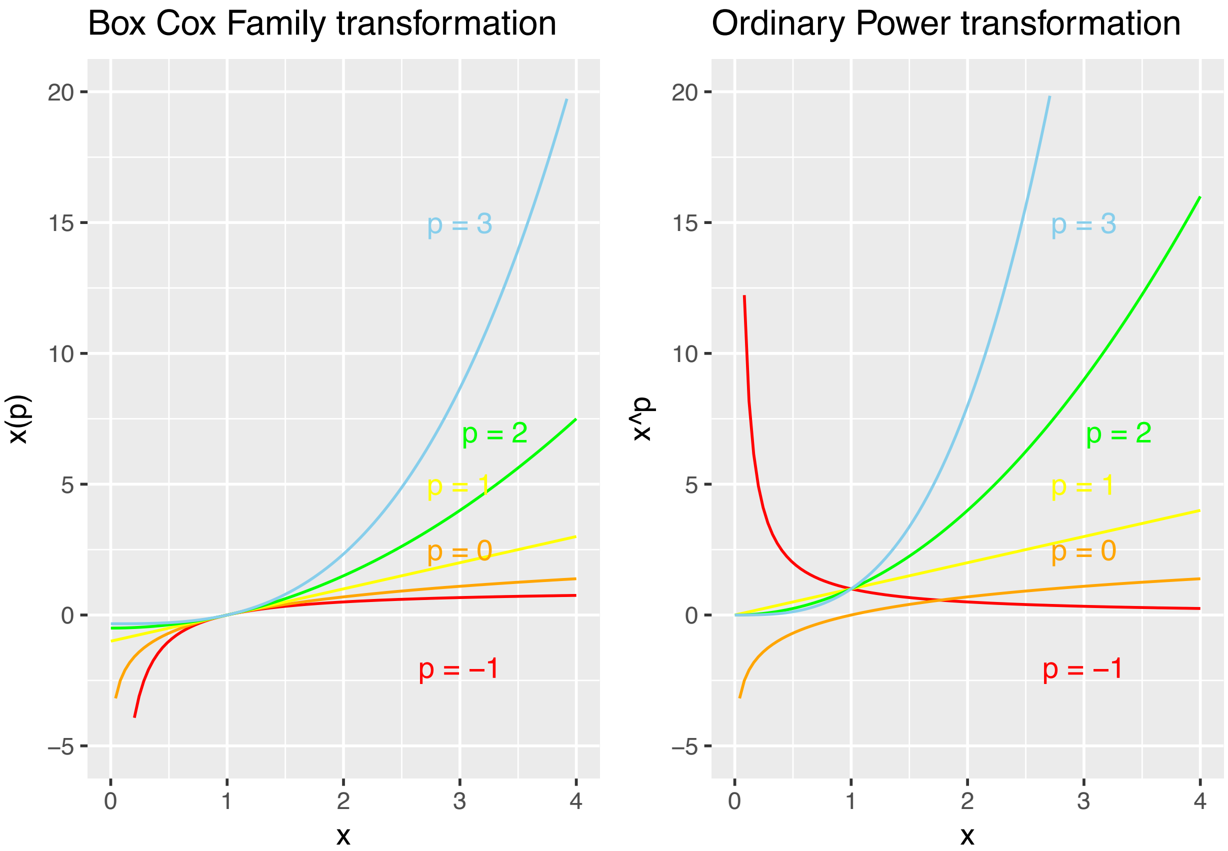
Exercise 4.1

```r
x = seq(0.001,4,length.out = 100);

box_cox_power_transformation = function(x, p){
  if(p == 0){
    return(log(x))
  } else {
    return((x^p - 1)/p)
  }
}
power_transformation = function(x, p){
  if(p == 0){
    return(log(x))
  } else {
    return(x^p)
  }
}


x_box_cox = data.frame(
  x = x,
  x_neg1 = box_cox_power_transformation(x, -1),
  x_0 = box_cox_power_transformation(x, 0),
  x_1 = box_cox_power_transformation(x, 1),
  x_2 = box_cox_power_transformation(x, 2),
  x_3 = box_cox_power_transformation(x, 3)
)

x_power = data.frame(
  x,
  power_neg1 = power_transformation(x, -1),
  power_x_0 =  power_transformation(x, 0),
  power_x_1 = power_transformation(x, 1),
  power_x_2 = power_transformation(x, 2),
  power_x_3 = power_transformation(x, 3)
)

p1 = ggplot() +
  geom_line(data = x_box_cox, aes(x=x, y=x_neg1), colour = "red") +
  geom_line(data = x_box_cox, aes(x=x, y=x_0), colour = "orange") +
  geom_line(data = x_box_cox, aes(x=x, y=x_1), colour = "yellow") +
  geom_line(data = x_box_cox, aes(x=x, y=x_2), colour = "green") +
  geom_line(data = x_box_cox, aes(x=x, y=x_3), colour = "skyblue") +
  xlab("x") +
  ylab("x(p)") +
  ggtitle("Box Cox Family transformation") +
  ylim(-5, 20) +
  annotate("text", label = "p = -1", colour = "red", x = 3, y = -2) +
  annotate("text", label = "p = 0", colour = "orange", x = 3, y = 2.5) +
  annotate("text", label = "p = 1", colour = "yellow", x = 3, y = 5) +
  annotate("text", label = "p = 2", colour = "green", x = 3.3, y = 7) +
  annotate("text", label = "p = 3", colour = "skyblue", x = 3, y = 15)



p2 = ggplot() +
  geom_line(data = x_power, aes(x=x, y=power_neg1), colour = "red") +
  geom_line(data = x_power, aes(x=x, y=power_x_0), colour = "orange") +
  geom_line(data = x_power, aes(x=x, y=power_x_1), colour = "yellow") +
  geom_line(data = x_power, aes(x=x, y=power_x_2), colour = "green") +
  geom_line(data = x_power, aes(x=x, y=power_x_3), colour = "skyblue") +
  xlab("x") +
  ylab("x^p") +
  ggtitle("Ordinary Power transformation") +
  ylim(-5, 20) +
  annotate("text", label = "p = -1", colour = "red", x = 3, y = -2) +
  annotate("text", label = "p = 0", colour = "orange", x = 3, y = 2.5) +
  annotate("text", label = "p = 1", colour = "yellow", x = 3, y = 5) +
  annotate("text", label = "p = 2", colour = "green", x = 3.3, y = 7) +
  annotate("text", label = "p = 3", colour = "skyblue", x = 3, y = 15)

pushViewport(viewport(layout = grid.layout(1, 2)))
print(p1, vp = viewport(layout.pos.row = 1, layout.pos.col = 1))
```

```
## Warning: Removed 5 rows containing missing values (geom_path).

## Warning: Removed 1 rows containing missing values (geom_path).

## Warning: Removed 2 rows containing missing values (geom_path).
```

```r
print(p2, vp = viewport(layout.pos.row = 1, layout.pos.col = 2))
```

```
## Warning: Removed 2 rows containing missing values (geom_path).

## Warning: Removed 1 rows containing missing values (geom_path).

## Warning: Removed 32 rows containing missing values (geom_path).
```



From the side-by-side report above, we can observe that
Similarities: The ranges for both box cox transformation and ordinary power are both (0, +inf). For $p > 0$ (where $p = 1, 2, 3$ in this graph), the output of both box cox transformation and ordinary power transformation both increase as x value increase. And as p value gets large, the more rapidly output values increase. For $p = 0$, the output of box cox transformation and ordinary power transformation are the same.

Differences: For $p < 0$ (where $p = -1$ in this graph), the output of box cox transformation increases as the input value gets larger. However, the output of ordinary transformation gets smaller as the input value gets larger.

Q2: $f(x) = \frac{(x^P - 1)}{P}$

$$f'(x) = \frac{P\cdot(Px^{P-1}) - (x^P - 1)\cdot 0}{P^2}$$

$$= \frac{P^2 x^{P-1} - 0}{P^2} = \frac{P^2 x^{P-1}}{P^2} = x^{P-1}$$

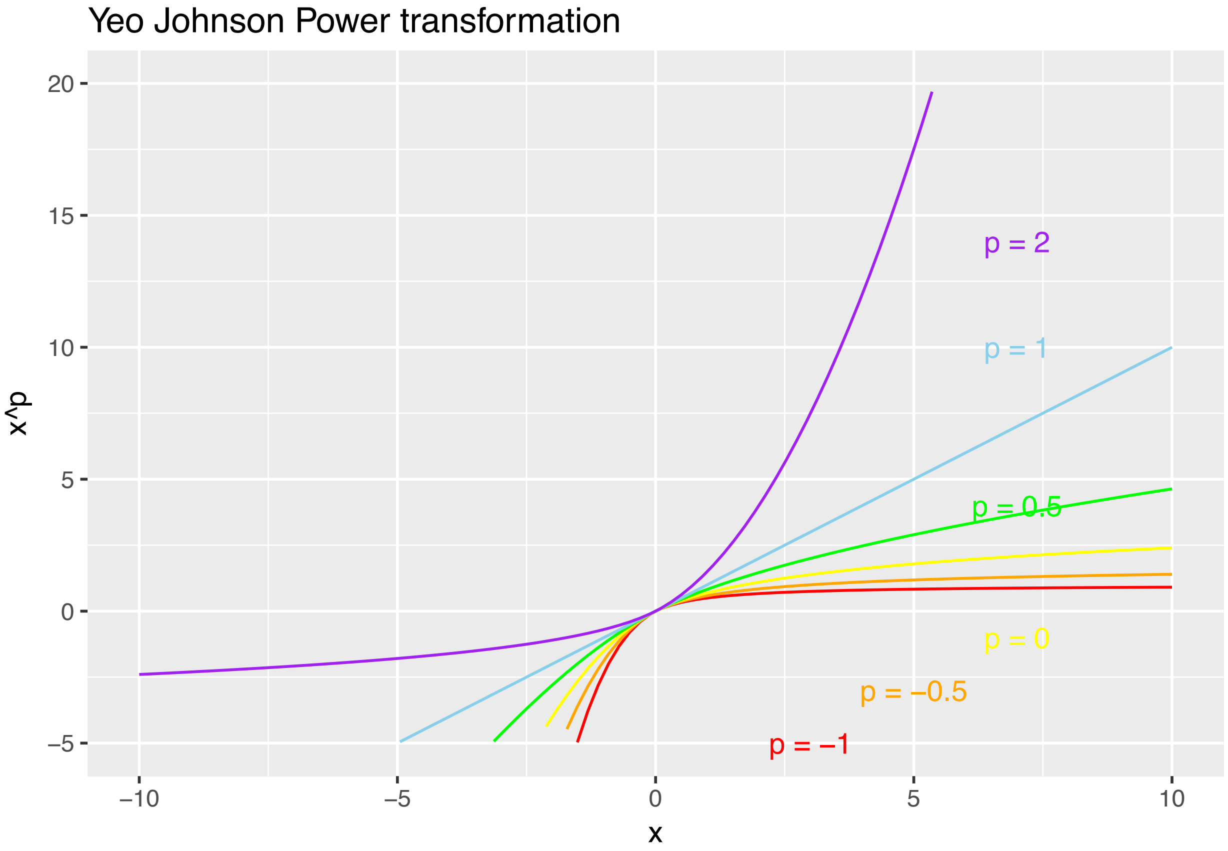
Thus if $x=1$, $f'(x) = x^{P-1} = 1^{P-1} = 1$ regardless of the value of P.

Exercise 4.4

Part a

```r
x = seq(-10, 10, length.out = 100)
yeo_johnson_transformation = function(x, p){
  y = rep(0, length(x))

  for(i in 1:length(x)){
    if(x[i] >= 0){
      y[i] = box_cox_power_transformation((x[i]+1), p)
    } else {
      y[i] = -1*box_cox_power_transformation((1-x[i]), (2-p))
    }
  }
  return(y)
}

yeo_trans_x_neg1 = yeo_johnson_transformation(x, -1)

yeo_transformed_x = data.frame(
  yeo_trans_x_neg1 = yeo_johnson_transformation(x, -1),
  yeo_trans_x_negpoint5 = yeo_johnson_transformation(x, -0.5),
  yeo_trans_x_0 = yeo_johnson_transformation(x, 0),
  yeo_trans_x_point5 = yeo_johnson_transformation(x, 0.5),
  yeo_trans_x_1 = yeo_johnson_transformation(x, 1),
  yeo_trans_x_2 = yeo_johnson_transformation(x, 2)
)

ggplot() +
  geom_line(data = yeo_transformed_x, aes(x=x, y=yeo_trans_x_neg1), colour = "red") +
  geom_line(data = yeo_transformed_x, aes(x=x, y=yeo_trans_x_negpoint5), colour = "orange") +
  geom_line(data = yeo_transformed_x, aes(x=x, y=yeo_trans_x_0), colour = "yellow") +
  geom_line(data = yeo_transformed_x, aes(x=x, y=yeo_trans_x_point5), colour = "green") +
  geom_line(data = yeo_transformed_x, aes(x=x, y=yeo_trans_x_1), colour = "skyblue") +
  geom_line(data = yeo_transformed_x, aes(x=x, y=yeo_trans_x_2), colour = "purple") +
  xlab("x") +
  ylab("x^p") +
  ggtitle("Yeo Johnson Power transformation") +
  ylim(-5, 20) +
  annotate("text", label = "p = -1", colour = "red", x = 3, y = -5) +
  annotate("text", label = "p = -0.5", colour = "orange", x = 5, y = -3) +
  annotate("text", label = "p = 0", colour = "yellow", x = 7, y = -1) +
  annotate("text", label = "p = 0.5", colour = "green", x = 7, y = 4) +
  annotate("text", label = "p = 1", colour = "skyblue", x = 7, y = 10) +
  annotate("text", label = "p = 2", colour = "purple", x = 7, y = 14)
```

```
## Warning: Removed 42 rows containing missing values (geom_path).

## Warning: Removed 41 rows containing missing values (geom_path).

## Warning: Removed 39 rows containing missing values (geom_path).

## Warning: Removed 34 rows containing missing values (geom_path).

## Warning: Removed 25 rows containing missing values (geom_path).
## Warning: Removed 23 rows containing missing values (geom_path).
```



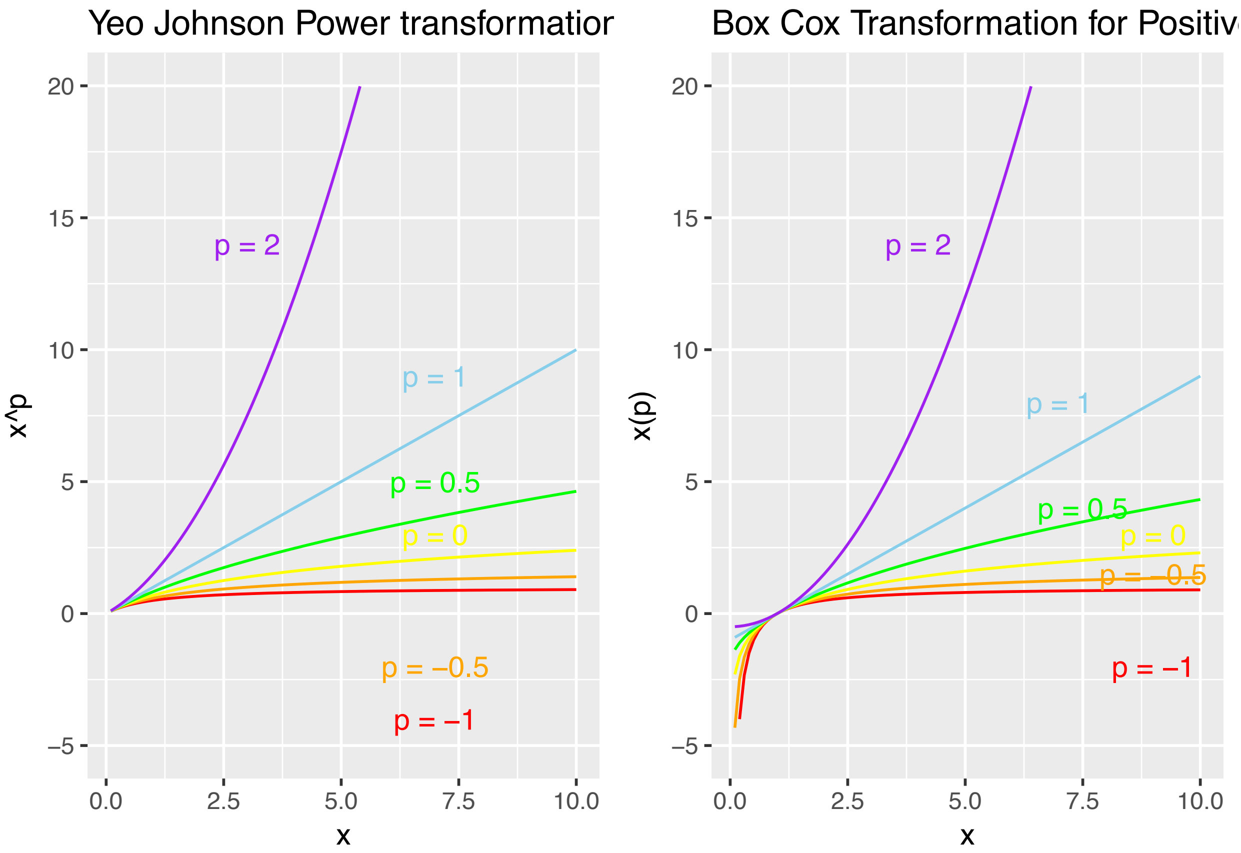
Part b

```
x = seq(0.1, 10, length.out = 100)

yeo_transformed_x = data.frame(
  yeo_trans_x_neg1 = yeo_johnson_transformation(x, -1),
  yeo_trans_x_negpoint5 = yeo_johnson_transformation(x, -0.5),
  yeo_trans_x_0 = yeo_johnson_transformation(x, 0),
  yeo_trans_x_point5 = yeo_johnson_transformation(x, 0.5),
  yeo_trans_x_1 = yeo_johnson_transformation(x, 1),
  yeo_trans_x_2 = yeo_johnson_transformation(x, 2)
)

p1 = ggplot() +
  geom_line(data = yeo_transformed_x, aes(x=x, y=yeo_trans_x_neg1), colour = "red") +
  geom_line(data = yeo_transformed_x, aes(x=x, y=yeo_trans_x_negpoint5), colour = "orange") +
  geom_line(data = yeo_transformed_x, aes(x=x, y=yeo_trans_x_0), colour = "yellow") +
  geom_line(data = yeo_transformed_x, aes(x=x, y=yeo_trans_x_point5), colour = "green") +
  geom_line(data = yeo_transformed_x, aes(x=x, y=yeo_trans_x_1), colour = "skyblue") +
  geom_line(data = yeo_transformed_x, aes(x=x, y=yeo_trans_x_2), colour = "purple") +
  xlab("x") +
  ylab("x^p") +
  ggtitle("Yeo Johnson Power transformation for Postive x") +
  ylim(-5, 20) +
  annotate("text", label = "p = -1", colour = "red", x = 7, y = -4) +
  annotate("text", label = "p = -0.5", colour = "orange", x = 7, y = -2) +
  annotate("text", label = "p = 0", colour = "yellow", x = 7, y = 3) +
  annotate("text", label = "p = 0.5", colour = "green", x = 7, y = 5) +
  annotate("text", label = "p = 1", colour = "skyblue", x = 7, y = 9) +
  annotate("text", label = "p = 2", colour = "purple", x = 3, y = 14)



x_box_cox = data.frame(
  x = x,
  x_neg1 = box_cox_power_transformation(x, -1),
  x_negpoint5 = box_cox_power_transformation(x, -0.5),
  x_0 = box_cox_power_transformation(x, 0),
  x_point5 = box_cox_power_transformation(x, 0.5),
  x_1 = box_cox_power_transformation(x, 1),
  x_2 = box_cox_power_transformation(x, 2)
)


p2 = ggplot() +
  geom_line(data = x_box_cox, aes(x=x, y=x_neg1), colour = "red") +
  geom_line(data = x_box_cox, aes(x=x, y=x_negpoint5), colour = "orange") +
  geom_line(data = x_box_cox, aes(x=x, y=x_0), colour = "yellow") +
  geom_line(data = x_box_cox, aes(x=x, y=x_point5), colour = "green") +
  geom_line(data = x_box_cox, aes(x=x, y=x_1), colour = "skyblue") +
  geom_line(data = x_box_cox, aes(x=x, y=x_2), colour = "purple") +
  xlab("x") +
  ylab("x(p)") +
  ggtitle("Box Cox Transformation for Positive x") +
  ylim(-5, 20) +
  annotate("text", label = "p = -1", colour = "red", x = 9, y = -2) +
  annotate("text", label = "p = -0.5", colour = "orange", x = 9, y = 1.5) +
  annotate("text", label = "p = 0", colour = "yellow", x = 9, y = 3) +
  annotate("text", label = "p = 0.5", colour = "green", x = 7.5, y = 4) +
  annotate("text", label = "p = 1", colour = "skyblue", x = 7, y = 8) +
  annotate("text", label = "p = 2", colour = "purple", x = 4, y = 14)

pushViewport(viewport(layout = grid.layout(1, 2)))
print(p1, vp = viewport(layout.pos.row = 1, layout.pos.col = 1))
```

```
## Warning: Removed 46 rows containing missing values (geom_path).
```

```
print(p2, vp = viewport(layout.pos.row = 1, layout.pos.col = 2))
```

```
## Warning: Removed 1 rows containing missing values (geom_path).

## Warning: Removed 36 rows containing missing values (geom_path).
```



From the side-by-side report above, we can observe that
The range for box cox transformation is strictly postive values, while Yeo Johnson Transformation can take any input value without strictly positive restrictions. Thus Yeo Johnson Transformation transforms all positive input values to positive values while Box Cox Transformation transforms all input values that are greater than 1 to positive output values and transforms input values that are strictly positive but less than 1 to negative output values. A around 0 input for Yeo Johnson transformation will produce 0 value as transformed output while a around 1 input for Box box transformation will produce 0 value as transformed output. The growth trend for them are the same.

Problem 2

Use the Baseball.xlsx uploaded in canvas.

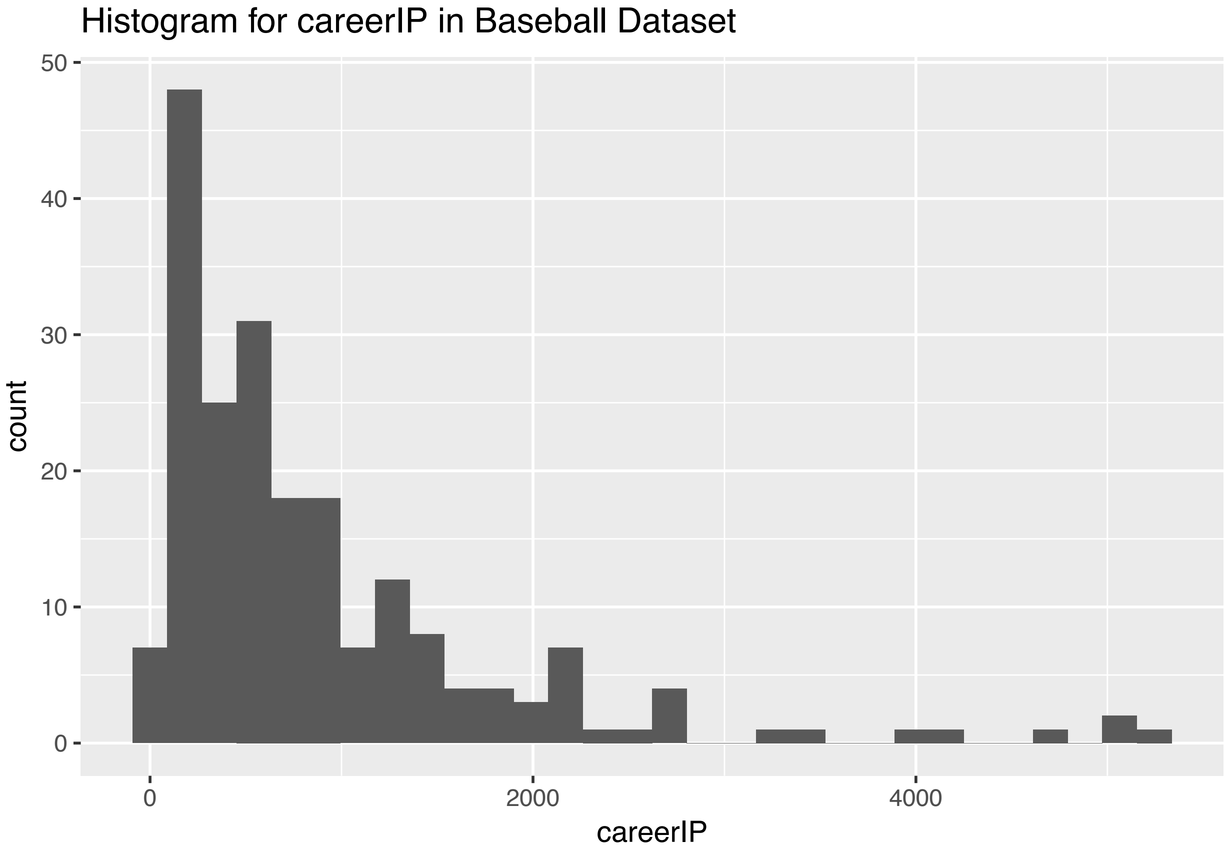
Part (a) Use the variable careerIP to create a histogram

```
# Loading the dataset baseball to RStudio
baseball= read_excel("~/Desktop/STAT 151A/STAT-151A/hw/hw2/baseball.xlsx")
str(baseball)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    206 obs. of  20 variables:
##  $ firstName: chr  "Don" "Jim" "Rick" "Doyle" ...
##  $ lastName : chr  "Aase" "Acker" "Aguilera" "Alexander" ...
##  $ team86   : chr  "Bal." "Atl." "N.Y." "Atl." ...
##  $ league86 : chr  "A" "N" "N" "N" ...
##  $ W86      : num  6 5 10 11 7 12 7 6 10 2 ...
##  $ L86      : num  7 12 7 10 2 7 3 10 10 4 ...
##  $ ERA86    : num  2.98 4.01 3.88 4.14 3.82 3.82 2.5 4.08 4.95 5.37 ...
##  $ G86      : num  66 44 28 34 22 28 61 60 62 36 ...
##  $ IP86     : num  81.2 155 141.2 228.1 113 ...
##  $ SV86     : num  34 0 0 0 0 1 7 10 7 5 ...
##  $ years    : num  9 4 2 16 8 11 1 4 1 3 ...
##  $ careerW  : num  61 20 20 160 53 122 7 19 10 4 ...
##  $ careerL  : num  54 20 14 135 58 108 3 28 10 7 ...
##  $ careerERA: num  3.74 3.99 3.58 3.71 3.65 3.49 2.5 3.97 4.95 4.35 ...
##  $ careerG  : num  325 175 49 467 367 369 61 202 62 60 ...
##  $ careerIP : num  956 411 264 2709 793 ...
##  $ careerSV : num  75 12 0 3 75 9 7 19 7 6 ...
##  $ salary   : chr  "625" "350" "195" "NA" ...
##  $ league87 : chr  "A" "N" "N" "N" ...
##  $ team87   : chr  "Bal." "Atl." "N.Y." "Atl." ...
```

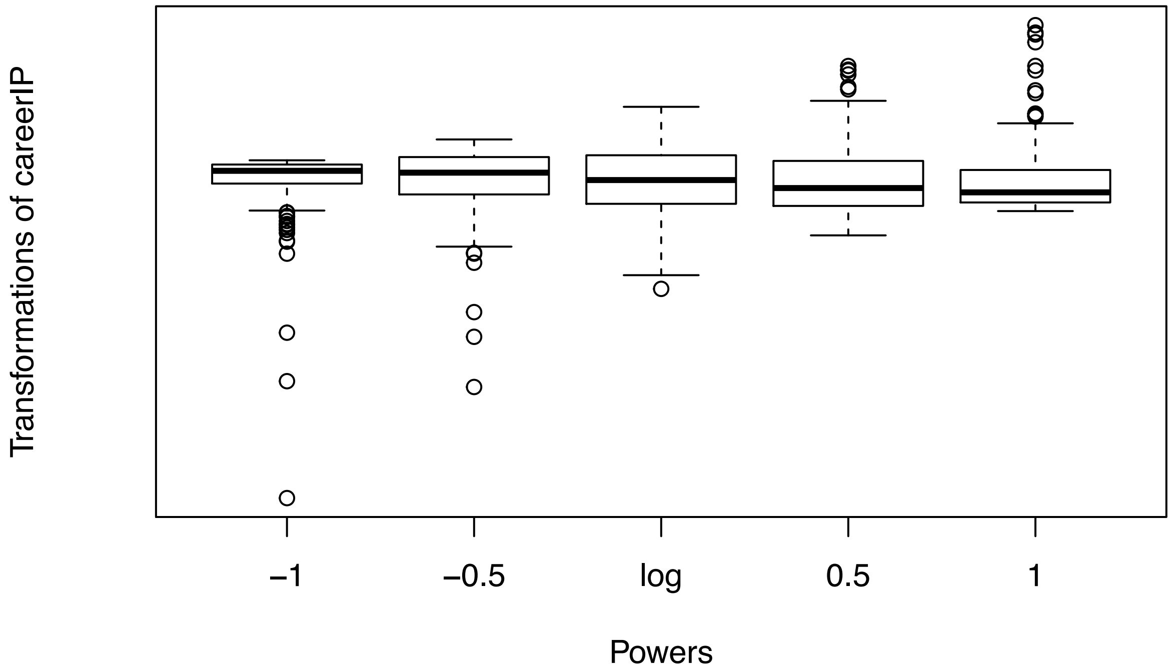
```r
ggplot(baseball, aes(x = careerIP)) +
  geom_histogram() +
  ggtitle("Histogram for careerIP in Baseball Dataset")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Part (b) Use symbox() function to predict the various transformation

```r
symbox(~careerIP, data = baseball)
```



From the graph we can see that log transformation produces the least outliers.

Part (c) Use the Hinge criterian for all the transformations that were displayed by symbox(). Which is the best transformation and why?

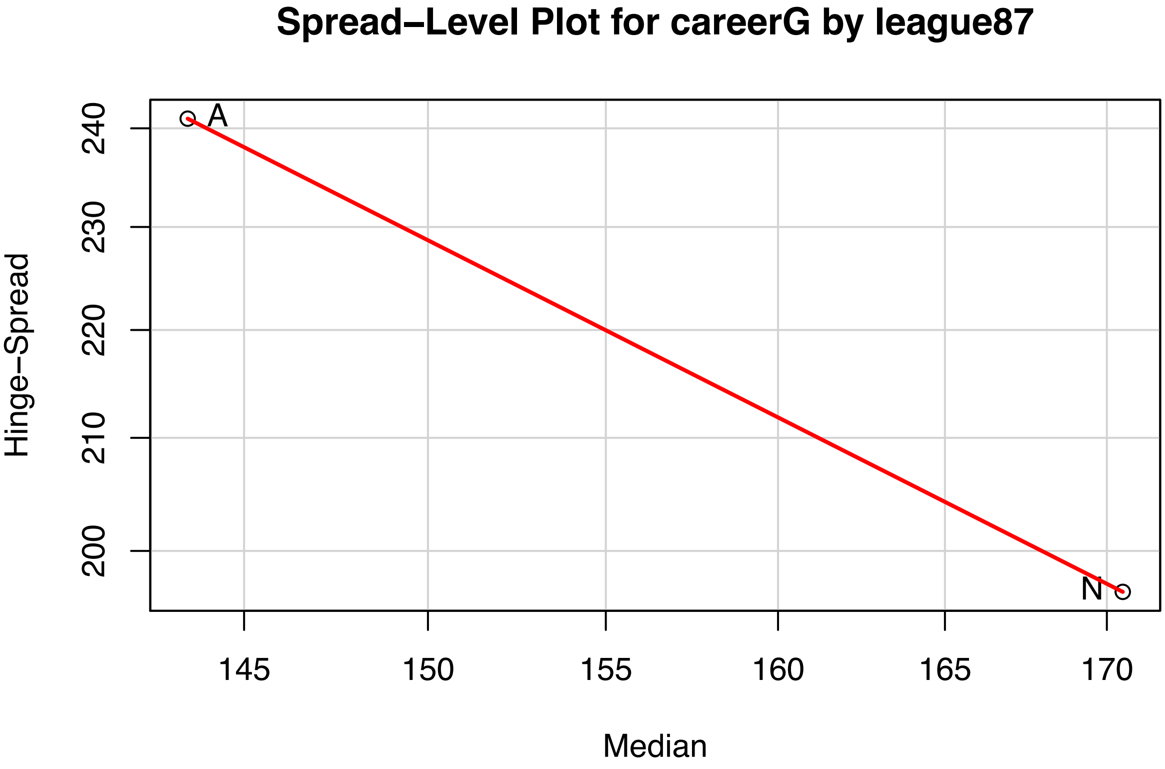
```
powers = c(-1, -0.5, 0, 0.5, 1)
careerIP = baseball$careerIP

hinge_ratio = function(data){

  upper_hinge = quantile(data, prob = 0.75)
  lower_hinge = quantile(data, prob = 0.25)
  median = quantile(data, prob = 0.5)
  return((upper_hinge - median)/(median - lower_hinge))
}

ratio = rep(0, 5)
for(i in 1:5) {
  if(powers[i] == 0){
    transformed_data = log(careerIP)
    ratio[i] = hinge_ratio(transformed_data)
  } else {
    transformed_data = careerIP^powers[i]
    ratio[i] = hinge_ratio(transformed_data)
  }
}
hinge_ratios = data.frame(
  powers = c(-1, -0.5, 0, 0.5, 1),
  ratio = ratio
)
hinge_ratios
```

```
##   powers    ratio
## 1   -1.0 2.028267
## 2   -0.5 1.397515
## 3    0.0 1.039439
## 4    0.5 1.511289
## 5    1.0 2.199298
```

Log Transformation is the best transformation for this dataset because we can see that when powers = 0 (log transformation), the hinge ratio is the closest to 1, which indicate the transformed data is the most symmetric.

Part (d) Create a plot between CareerG grouped by league use the spreadLevelPlot() to predict the transformation. Try the suggested transformation. Give your rationale why it works

```
spreadLevelPlot(careerG~league87, baseball)
```

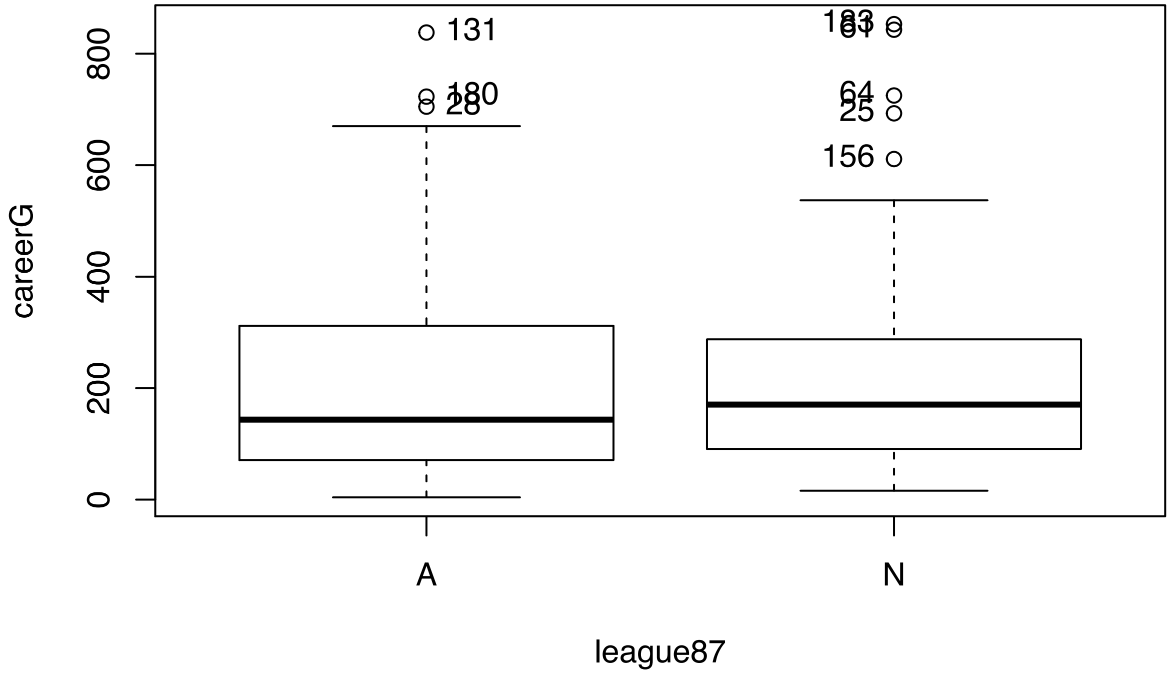


```
##   LowerHinge Median UpperHinge Hinge-Spread
## A         71  143.5      312.0        241.0
## N         91  170.5      287.5        196.5
##
## Suggested power transformation:  2.184073
```

Thus we can know that the suggested power transformation is 2.184073.

```
# get transformed careerG dataset
careerG_transformed = box_cox_power_transformation(baseball$careerG, 2.184)

# plot boxplot for original and transformed dataset
Boxplot(careerG~league87, baseball)
```

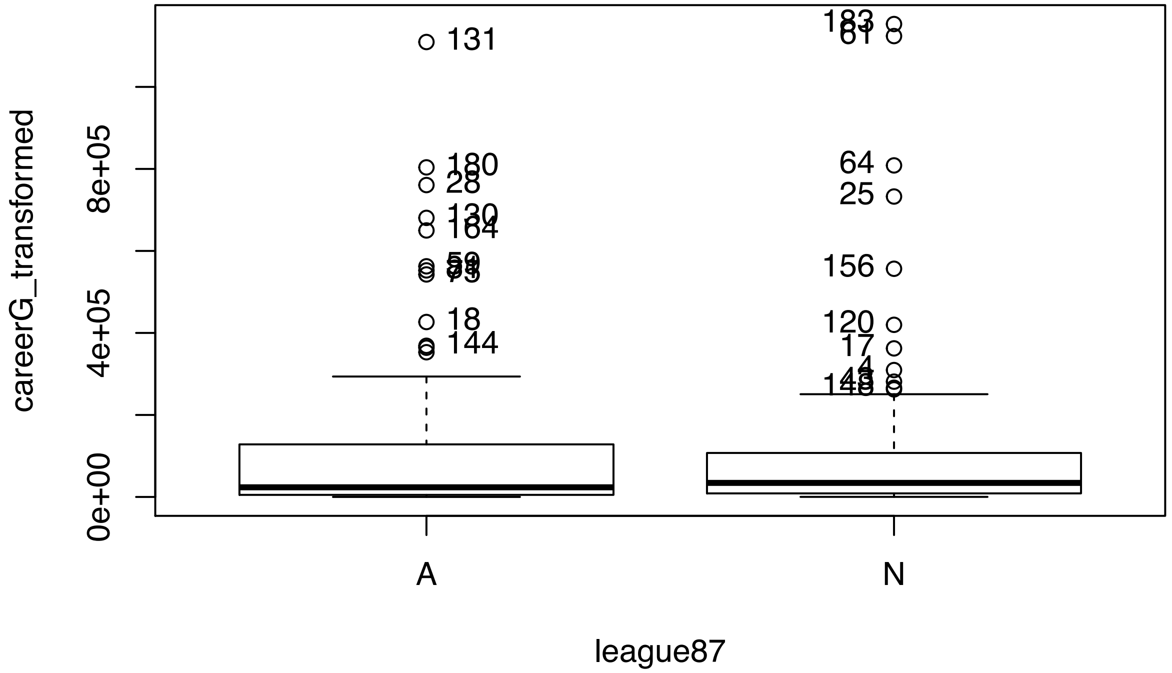
```
## Warning in Boxplot.default(mf[[response]], x, labels = mf[[lab.var]], xlab
## = xlab, : NAs introduced by coercion
```



```
## [1] "28"  "131" "180" "25"  "61"  "64"  "156" "183"
```

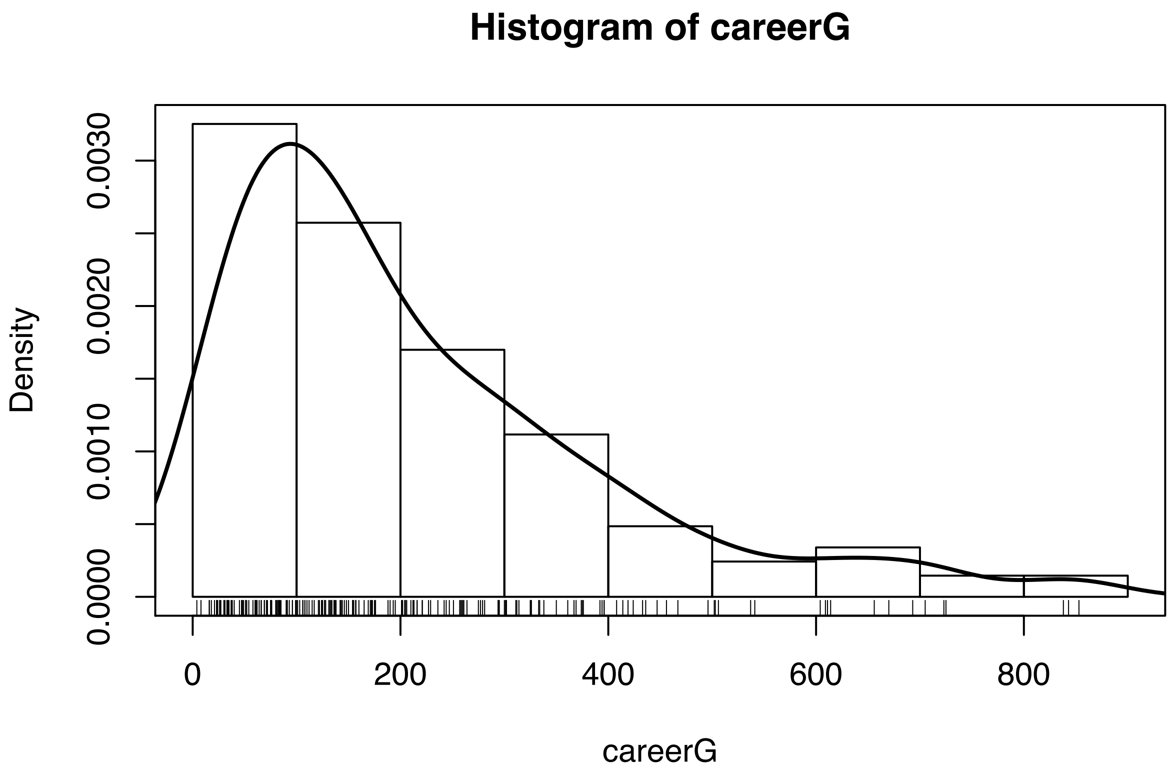
```
Boxplot(careerG_transformed~league87, data=baseball)
```

```
## Warning in Boxplot.default(mf[[response]], x, labels = mf[[lab.var]], xlab
## = xlab, : NAs introduced by coercion
```

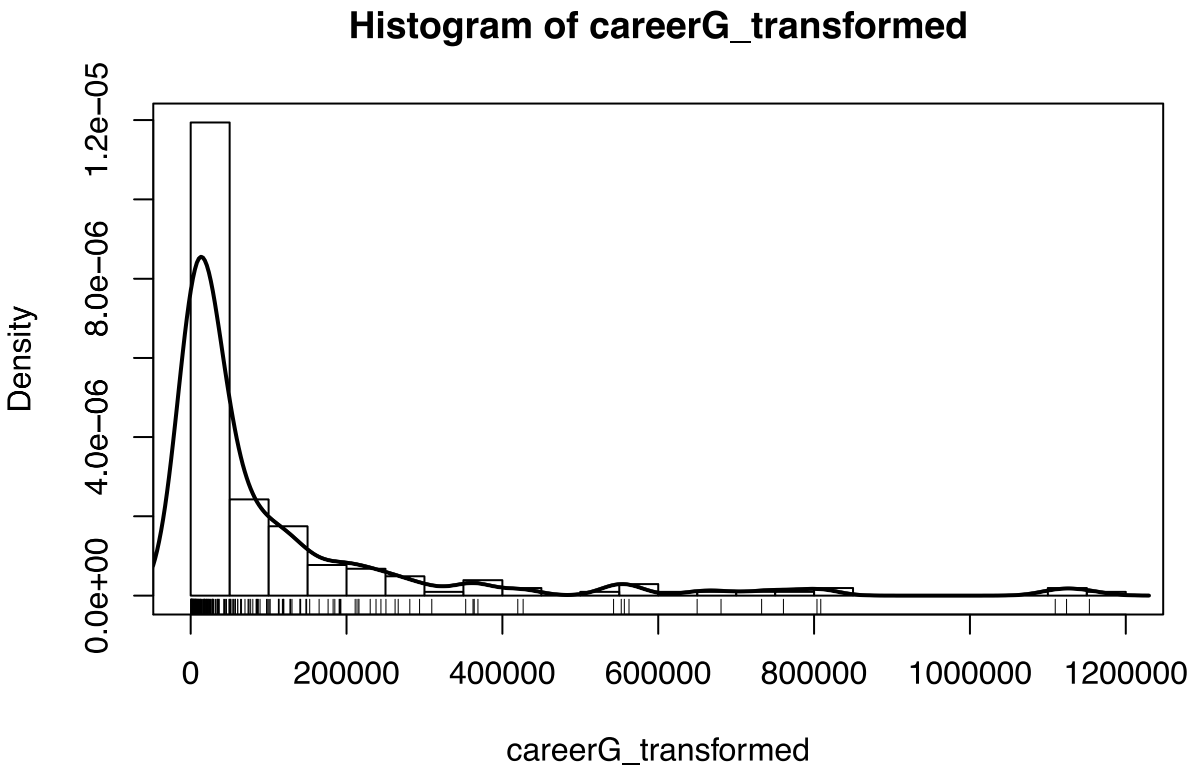


```
##  [1] "131" "180" "28"  "130" "164" "59"  "81"  "75"  "18"  "144" "183"
## [12] "61"  "64"  "25"  "156" "120" "17"  "4"   "43"  "148"
```

```
# plot density plot for original dataset
with(baseball, {
  hist(careerG, breaks = "FD", freq = FALSE, ylab = "Density")
  lines(density(careerG), lwd = 2)
  rug(careerG)
  box()
})
```

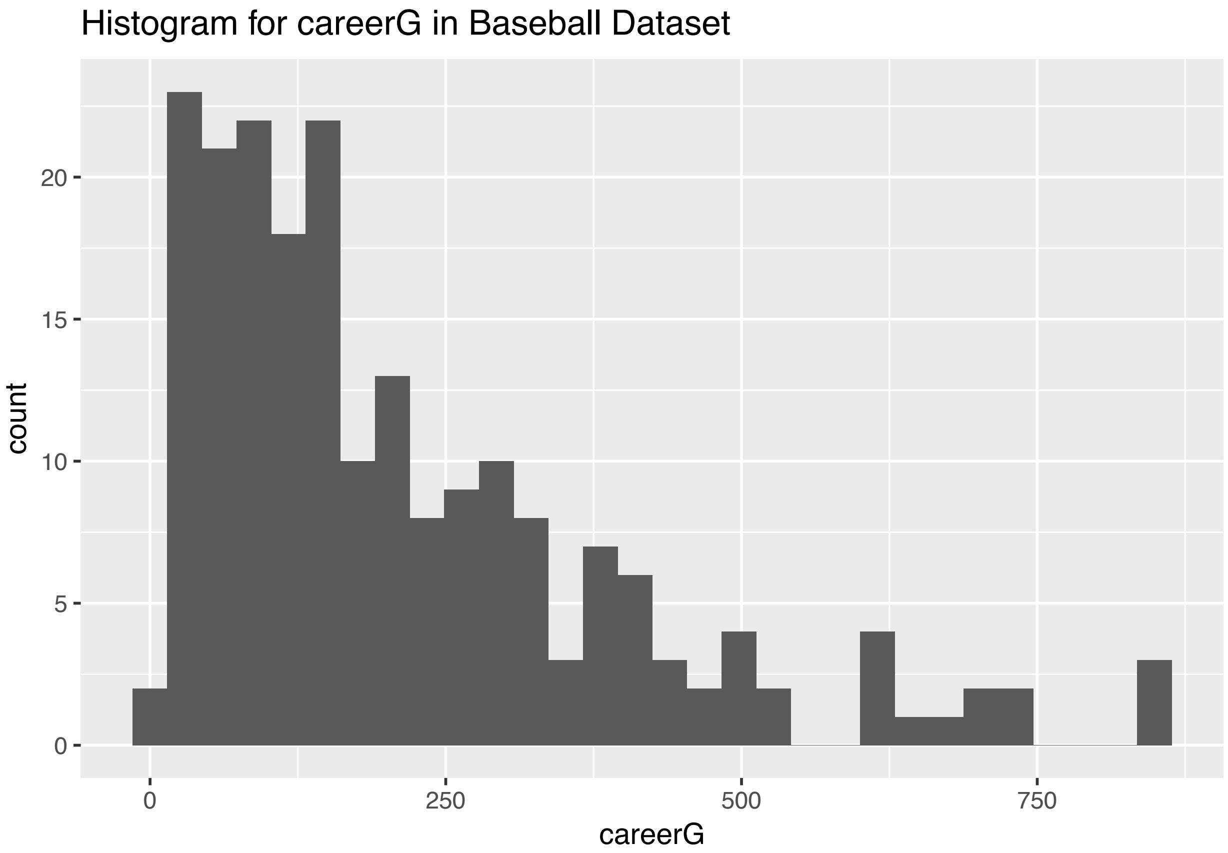


```
# plot density plot for transformed dataset
with(baseball, {
  hist(careerG_transformed, breaks = "FD", freq = FALSE, ylab = "Density")
  lines(density(careerG_transformed), lwd = 2)
  rug(careerG_transformed)
  box()
})
```



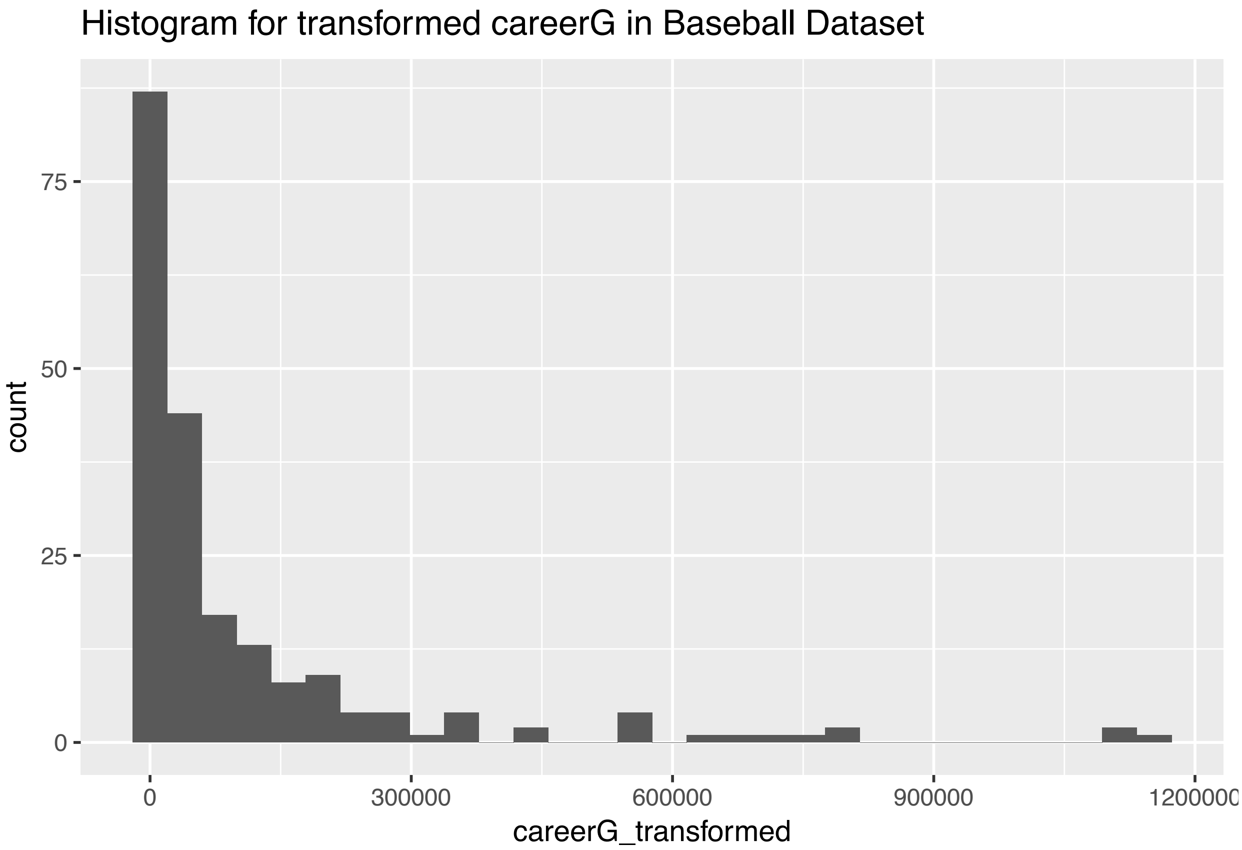
```r
# plot histogram for original dataset
ggplot(baseball, aes(x = careerG)) +
  geom_histogram() +
  ggtitle("Histogram for careerG in Baseball Dataset")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# plot histogram for transformed dataset
ggplot(baseball, aes(x = careerG_transformed)) +
  geom_histogram() +
  ggtitle("Histogram for transformed careerG in Baseball Dataset")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



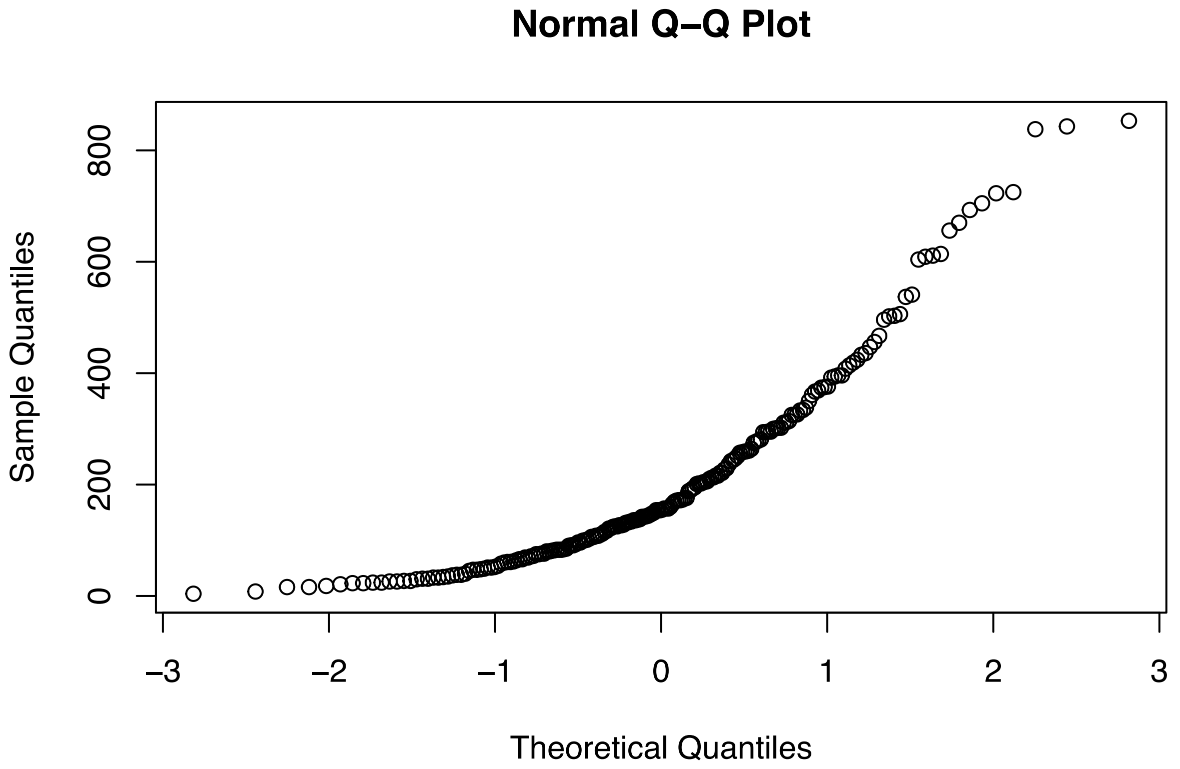
```
r1 = hinge_ratio(careerG_transformed)
r2 = hinge_ratio(baseball$careerG)

r1
```

```
##      75%
## 4.234288
```

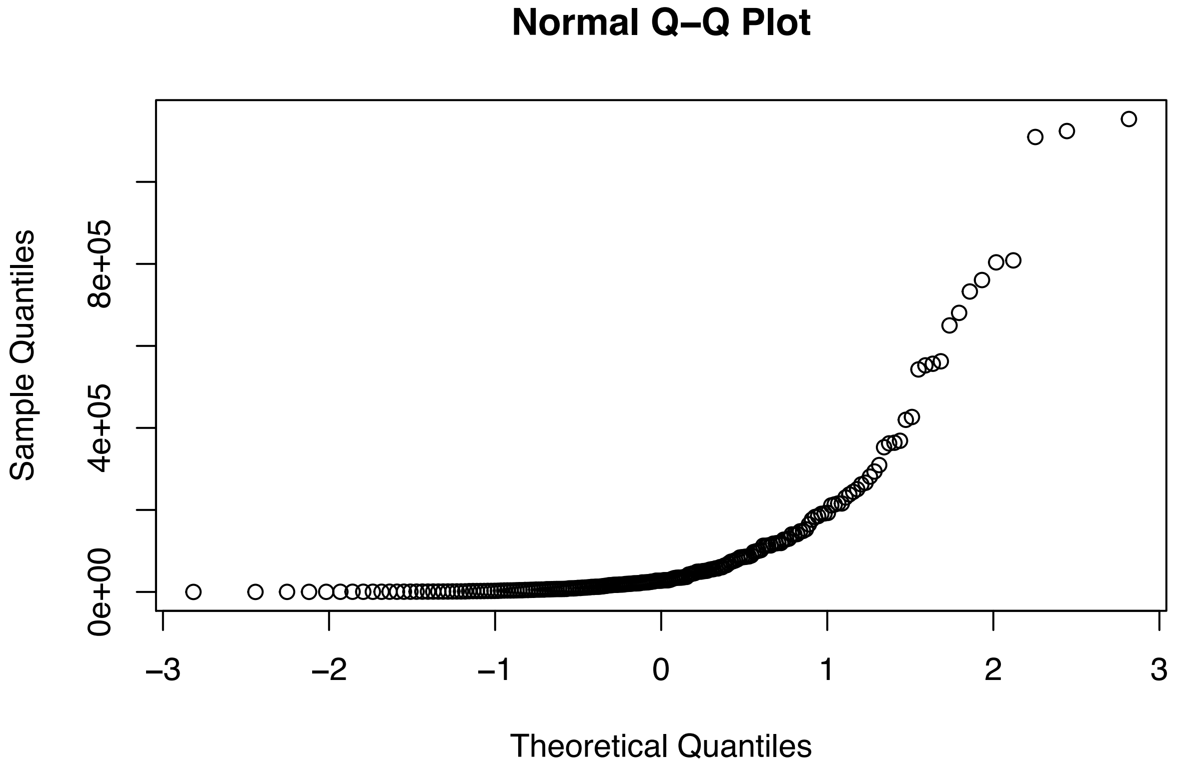
```
r2
```

```
##      75%
## 1.942761
```

```
qqnorm(baseball$careerG)
```



```r
qqnorm(careerG_transformed)
```



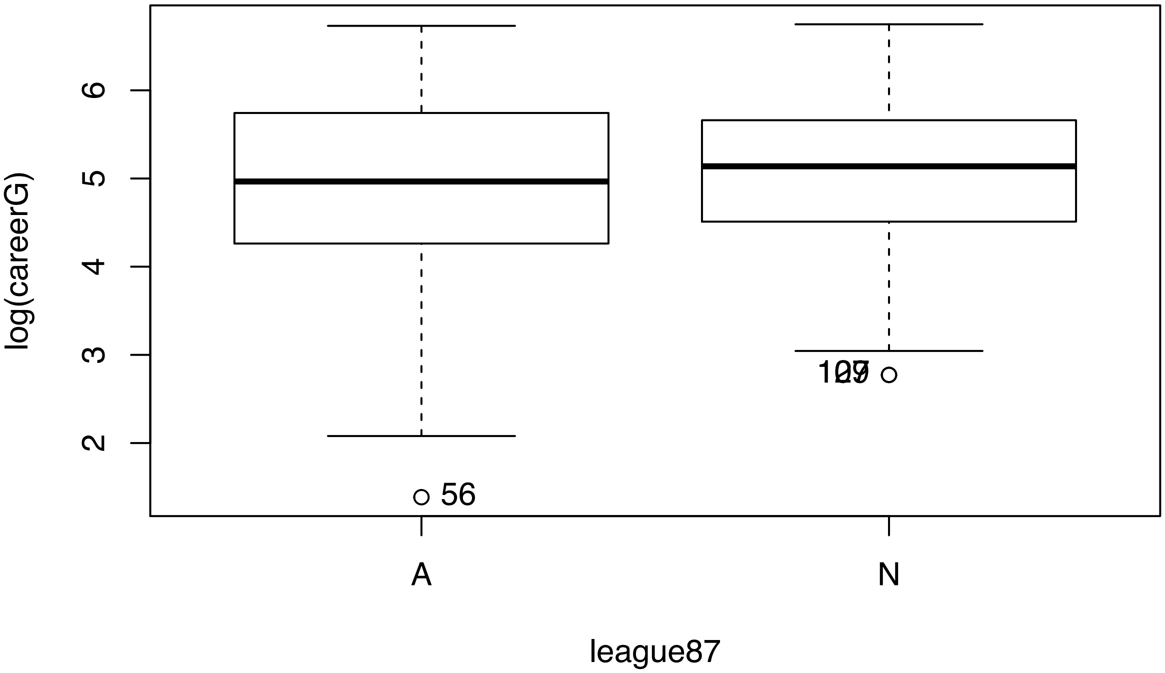
As we can see from the decreasing hinge ratio of power transformed data with p = 2.184, the transformed data is tend to be symmetric after transformation.

The reasoning behind the suggested power transformation might because the original dataset is heavy-tailed and power transform with p = 2 transform the dataset to right skewed. However, from the histogram and density plot of original data, we can see that the original data is also high right skewed. Thus, the textbook suggests that we should use power transformation with small p since descending the ladder of powers (e.g., to logX) tends to correct a positive skew.

Thus, I try to do the log transformation below, plot related plots and calculated it hinge ratio. Here, we can see that the result dataset is slightly left skewed and its hinge value is very close to 1. Since p = 0 transformation overly transform the dataset to left skewed, it seems that a positive small value of p is a good candidate for the transformation of this dataset.

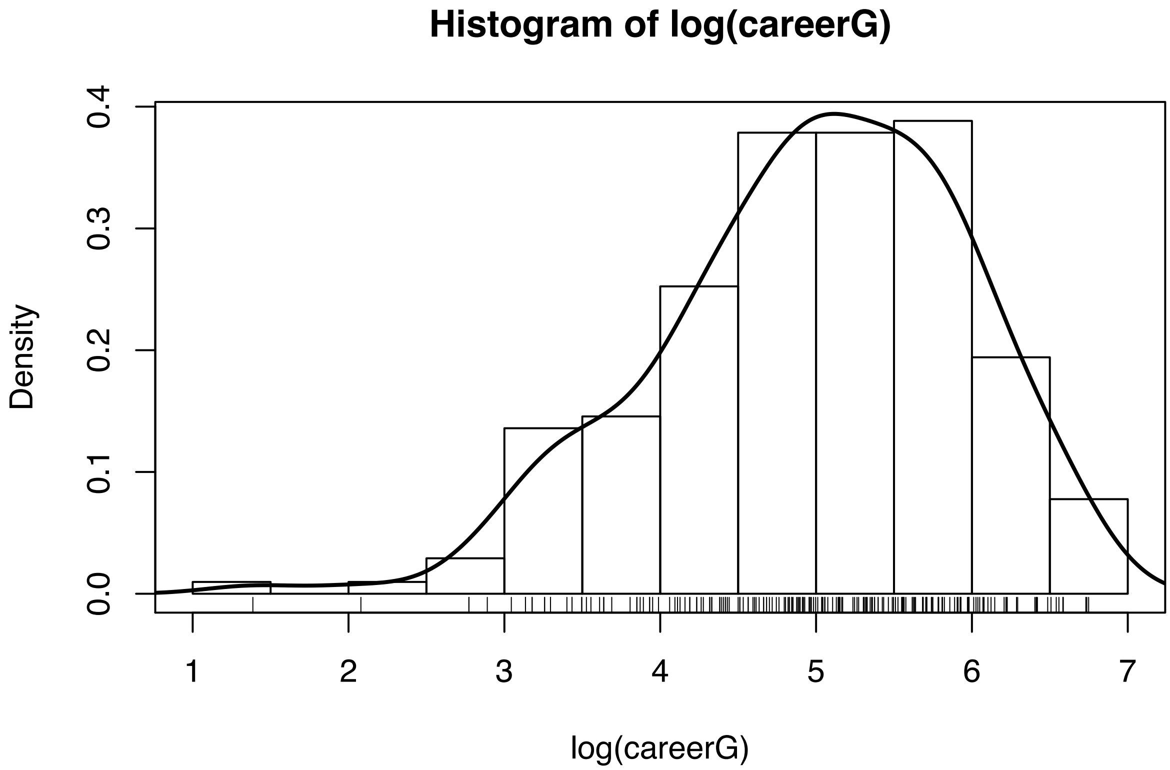
```r
Boxplot(log(careerG)~league87, data=baseball)
```

```
## Warning in Boxplot.default(mf[[response]], x, labels = mf[[lab.var]], xlab
## = xlab, : NAs introduced by coercion
```



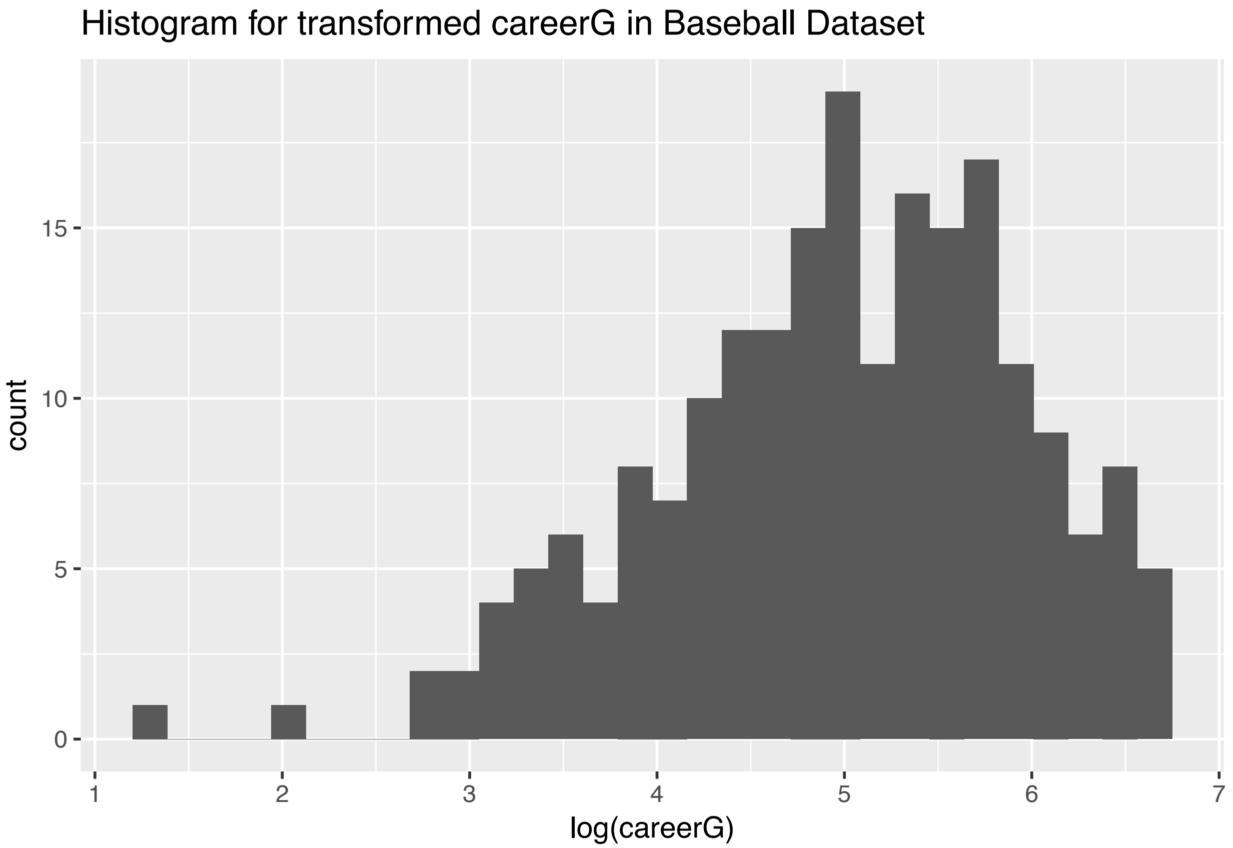
```
## [1] "56"  "109" "127"
```

```r
# plot density plot for original dataset
with(baseball, {
  hist(log(careerG), breaks = "FD", freq = FALSE, ylab = "Density")
  lines(density(log(careerG)), lwd = 2)
  rug(log(careerG))
  box()
})
```



```
ggplot(baseball, aes(x = log(careerG))) +
  geom_histogram() +
  ggtitle("Histogram for transformed careerG in Baseball Dataset")
```

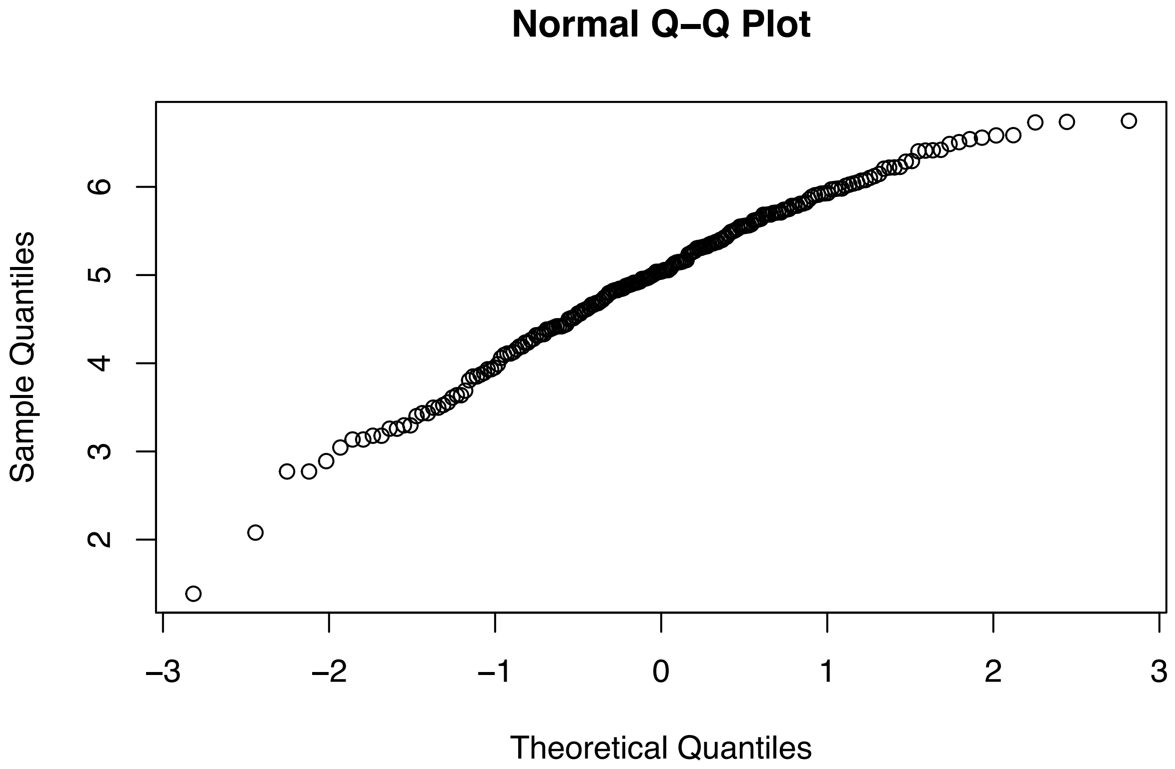
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
r3 = hinge_ratio(log(baseball$careerG))
r3
```

```
##      75%
## 1.006618
```

```
qqnorm(log(baseball$careerG))
```


Normal Q–Q Plot
Sample Quantiles
2
3
4
5
6
Theoretical Quantiles
−3
−2
−1
0
1
2
3