# CAssignment

Oksana Harapyn

2024-06-30

## Question 1

### Analysis of Variable x.n

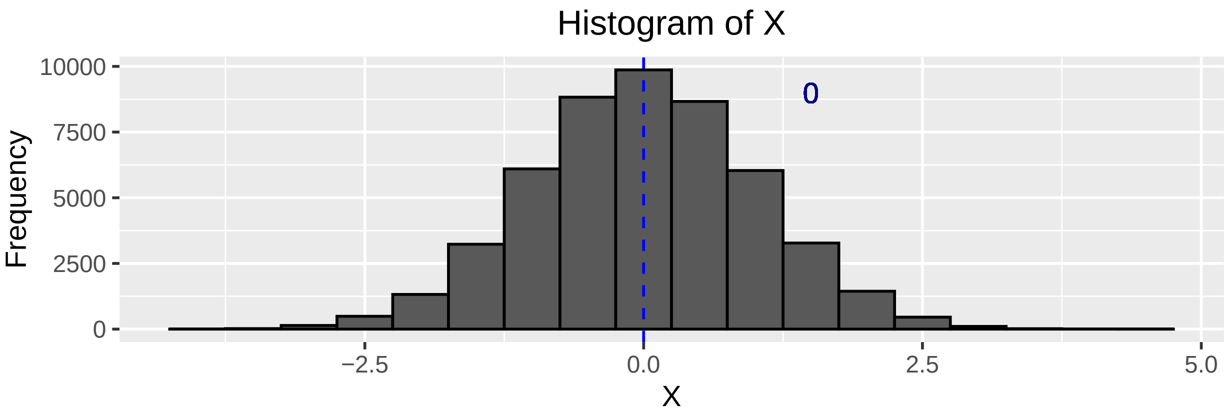
The histogram has symmetric bell shape, where most of the values are centered around expected value 0 and variance 1. It is to be expected since the sample taken for observation is rather big and rnorm function generated the sample with mean and standard deviation close to 0 and 1 respectfully.

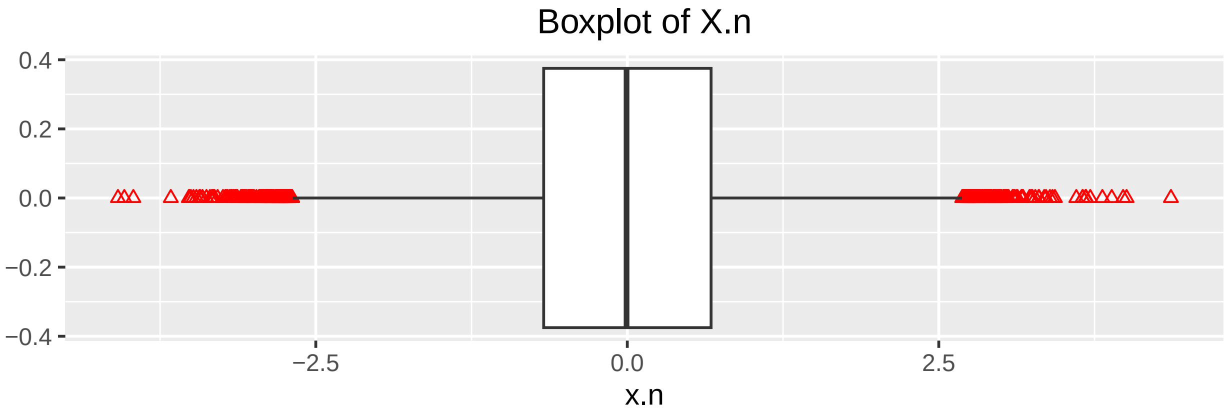
Mean and standard deviation can be helpful in variable's distribution evaluation. For example, since our distribution of x.n is normal, we can reasonably expect for most of the values to be in range

$$-2; 2$$

. However, cautious consideration of expected value and standart deviation are still required, since neither of those statistics represents true nature of the data. Mean and standard deviation may be biased by the outliers or skewness, distribution may be bi modal so that separation groups have to be found first, etc.

```r
set.seed(100)
Data=data.frame(x.n=rnorm(50000),x.p=rPareto(50000,t=1,alpha=2))

p1= ggplot(Data, aes(x=x.n))+
  geom_histogram(color="black", binwidth=0.5)+
  geom_vline(aes(xintercept=mean(x.n)), color="blue", linetype="dashed")+
  ggtitle("Histogram of X")+
  xlab("X")+
  ylab("Frequency")+
  theme(plot.title = element_text(hjust = 0.5))+
  geom_text(x=1.5,y=9000, label="0", color="darkblue",size=4)

p2= ggplot(Data, mapping=(aes(x=x.n)))+
geom_boxplot(outlier.shape =2, outlier.colour ="red")+
ggtitle("Boxplot of X.n")+
theme(plot.title = element_text(hjust = 0.5))

grid.arrange(p1,p2, nrow=2)
```





```
print(paste("Mean of X.n:", mean(Data$x.n)))
```

```
## [1] "Mean of X.n: -0.000208495582248687"
```

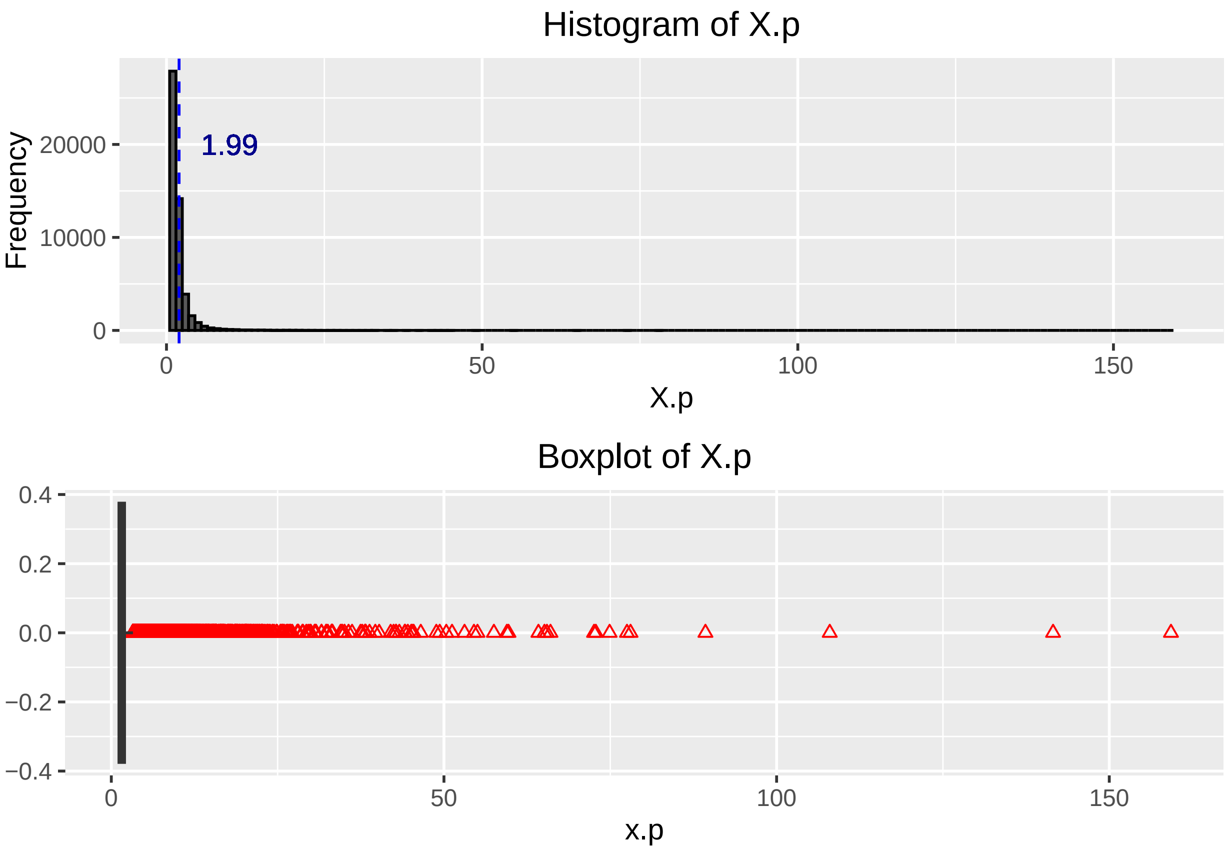
```
print(paste("Standart Deviation of X.n:", sd(Data$x.n)))
```

```
## [1] "Standart Deviation of X.n: 0.998965809425168"
```

### Analysis of Variable x.p

The statement provided for x.p seem trustworthy. Boxplot of x.p shows the great amount of extreme values in the data. Even through most of the variables in data seem to be around 1, outliers with considerably greater values shift the mean line to the right. Since the mean is biased by the outliers, value of standard deviation is also less relevant for summarizing and predicting the data. If we use the IQR method to sort the outliers out, values of expected value and standard deviation will decrease. However, before sorting the extreme values out, the additional consideration needed, so we wouldn't transform the data too much and change the pattern

```
p3= ggplot(Data, aes(x=x.p))+
  geom_histogram(color="black", binwidth=1)+
  geom_vline(aes(xintercept=mean(x.p)), color="blue", linetype="dashed")+
  ggtitle("Histogram of X.p")+
  xlab("X.p")+
  ylab("Frequency")+
  theme(plot.title = element_text(hjust = 0.5))+
  geom_text(x=10,y=20000, label="1.99", color="darkblue",size=4)

p4= ggplot(Data, mapping=(aes(x=x.p)))+
geom_boxplot(outlier.shape =2, outlier.colour ="red")+
ggtitle("Boxplot of X.p")+
theme(plot.title = element_text(hjust = 0.5))

grid.arrange(p3,p4, nrow=2)
```



```
m1=mean(Data$x.p)
sd1=sd(Data$x.p)

print(paste("Mean of X.p:", m1))
```

```
## [1] "Mean of X.p: 1.99390436133807"
```

```
print(paste("Standart Deviation of X.p:", sd1))
```

```
## [1] "Standart Deviation of X.p: 2.6011734070772"
```

```
IQR=quantile(Data$x.p, 0.75)-quantile(Data$x.p, 0.25)
newData=filter(Data, Data$x.p <= quantile(Data$x.p, 0.75) +3*IQR)
m2=mean(newData$x.p)
```

```
sd2=sd(newData$x.p)
print(paste("Mean of X.p without Outliers:", m2))
```

```
## [1] "Mean of X.p without Outliers: 1.63370937387356"
```

```
print(paste("Standart Deviation of X.p without Outliers:", sd2))
```

```
## [1] "Standart Deviation of X.p without Outliers: 0.695393325657267"
```

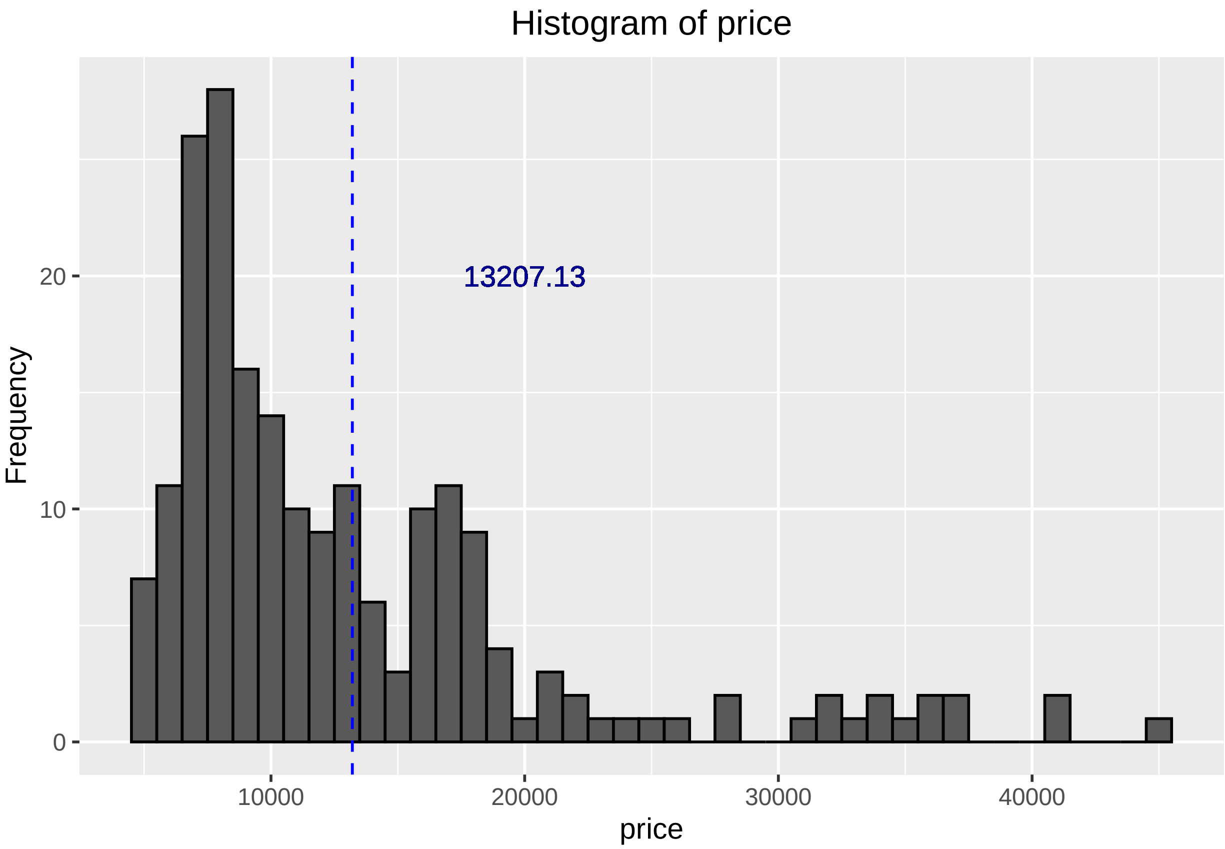
## Question 2

### Histogram and Mean of Variable price

```
Data = read.csv("Car_data.csv", na.strings=c("?", "NA", "na", "N/A"))
Data=Data[-which(is.na(Data$price)), ]
print(paste("Mean of price:", mean(Data$price)))
```

```
## [1] "Mean of price: 13207.1293532338"
```

```
ggplot(Data, aes(x=price))+
  geom_histogram(color="black", binwidth=1000)+
  geom_vline(aes(xintercept=mean(price)), color="blue", linetype="dashed")+
  ggtitle("Histogram of price")+
  xlab("price")+
  ylab("Frequency")+
  theme(plot.title = element_text(hjust = 0.5))+
  geom_text(x=20000,y=20, label="13207.13", color="darkblue",size=4)
```

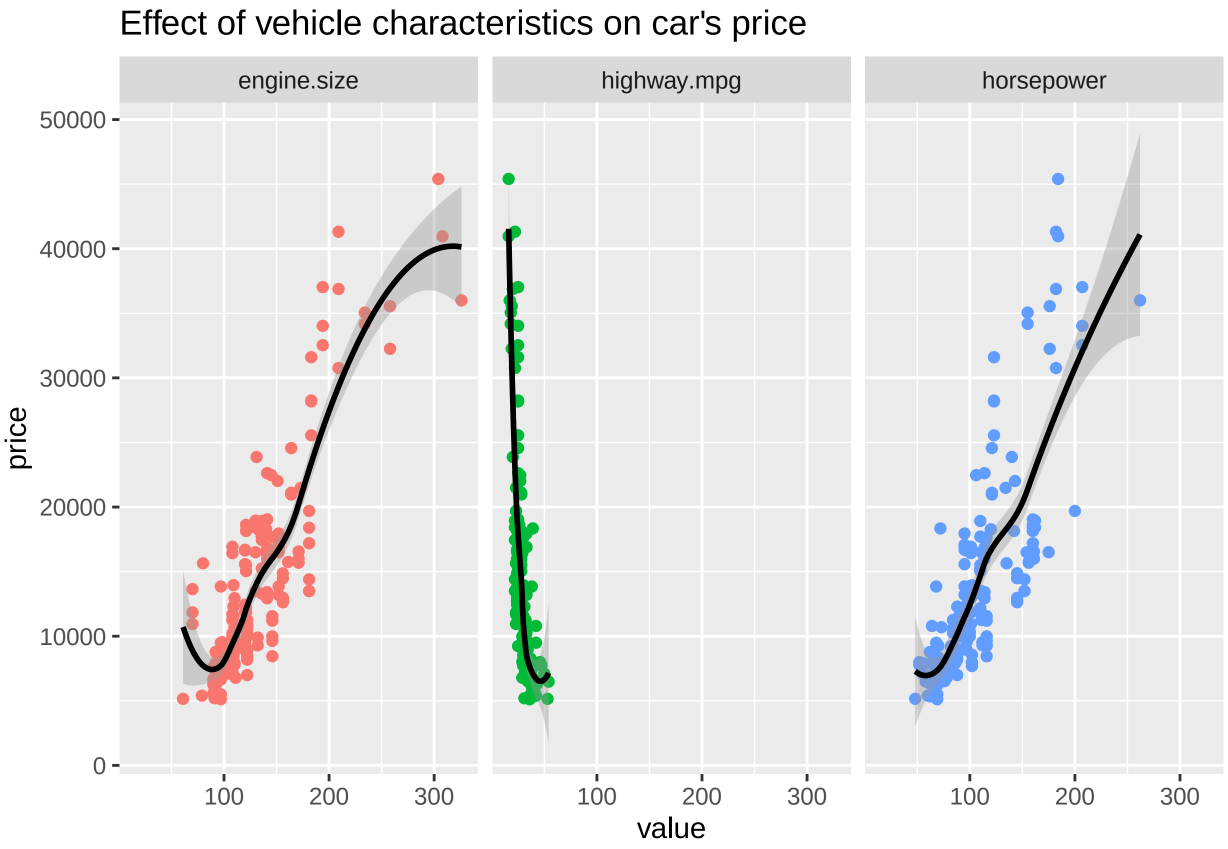


### Relation between variables price,curb.weight, engine.size, horsepower, highway.mpg and price Considering the scatter plots we may say that in general the greater the value of the variables engine.size, horsepower and curb.weight, the more pricey is the vehicle. The opposite works for highway.mpg. With the rise of those 4 car characteristics, the greater spread of the price for each of them is.

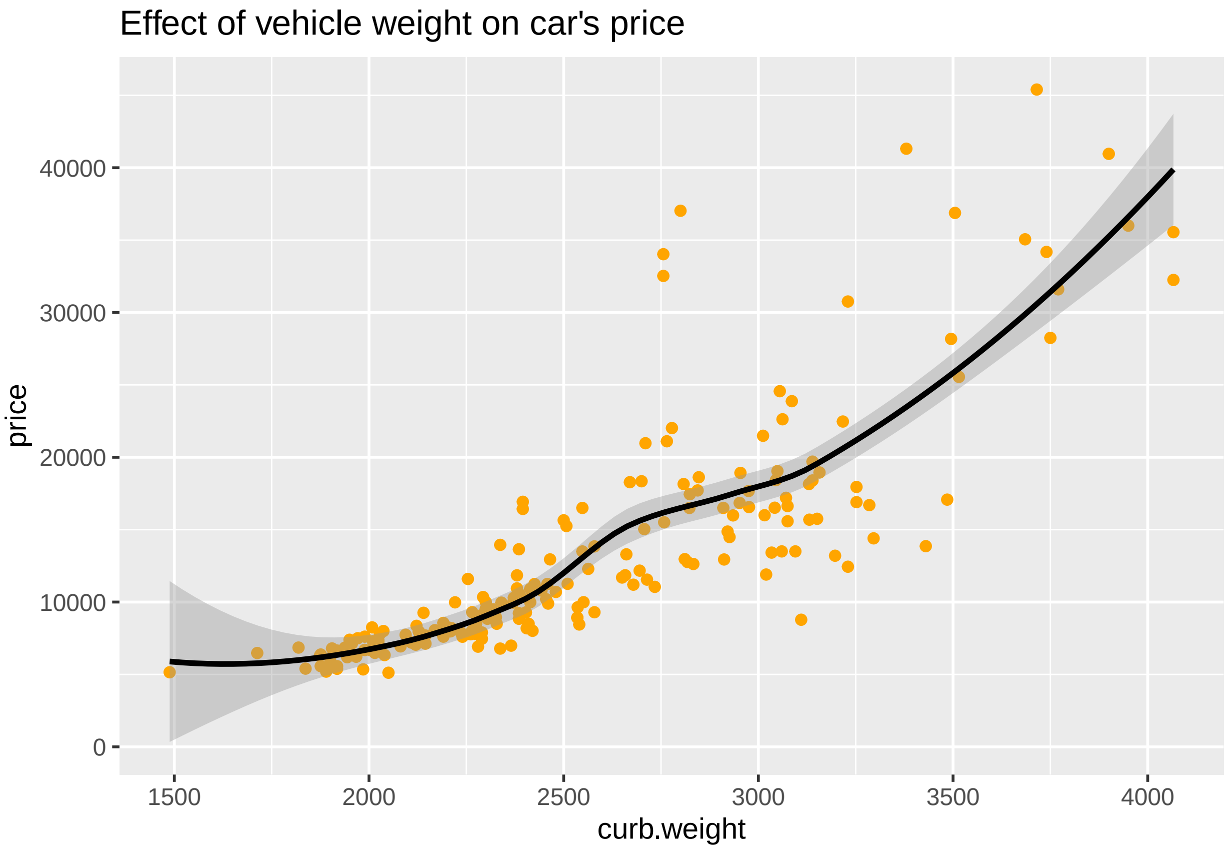
```r
shortData = dplyr::select(Data, c(price,curb.weight, engine.size, horsepower, highway.mpg))
shortData=shortData[-which(is.na(Data$horsepower)), ]

shortData1 = dplyr::select(Data, c(price, engine.size, horsepower, highway.mpg))
gatheredData=shortData1 %>%
  as_tibble() %>%
  gather(key = "variable", value = "value",
         -price)
p5=ggplot(gatheredData, aes(x = value, y = price)) +
  facet_wrap(~variable)+
  geom_point(aes(color=variable))+theme(legend.position = "none")+
  ggtitle("Effect of vehicle characteristics on car's price")+
  stat_smooth(formula = y ~ x, method = "loess", color="black")

shortData2 = dplyr::select(Data, c(price,curb.weight))
p6=ggplot(shortData2, aes(x=curb.weight, y=price))+
  geom_point(color="orange")+ggtitle("Effect of vehicle weight on car's price")+
  stat_smooth(formula = y ~ x, method = "loess", color="black")

p5
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## ('stat_smooth()').
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## ('geom_point()').
```



```
p6
```



## PCA analysis

Proportions of the PCA suggests that we can explain 99% of the variance using first tree principal components.

```
shortData=scale(shortData[,-1])
#round(cov(shortData), digits=3)
shortData.PCA = prcomp(shortData)
PC=shortData.PCA$rotation
round(PC,digits=3)
```

```
##                PC1    PC2    PC3    PC4
## curb.weight  0.507 -0.186 -0.657 -0.526
## engine.size  0.500 -0.636  0.103  0.578
## horsepower   0.505  0.100  0.726 -0.456
## highway.mpg -0.488 -0.742  0.173 -0.425
```

```
summary(shortData.PCA)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4
## Standard deviation     1.8318 0.57125 0.48861 0.28209
## Proportion of Variance 0.8388 0.08158 0.05969 0.01989
## Cumulative Proportion  0.8388 0.92042 0.98011 1.00000
```

PC1: First principal component holds most of the information. It reflects car's capacity. All four variables have an equally medium effect on it, aside from the fact that highway.mpg has it negative. The more powerful the engine and the bigger a vehicle is, the more fuel it needs to operate.

PC2: Second principal component reflects car's fuel burn rate. While curb.weight and horsepower have insignificant effect on it, engine.size and highway.mpg have strong negative influence on this value. The smaller engineer of the car burn more gallon per mile and therefore car is less budget friendly.

PC3: the more gallons car require per mile, the more fuel the car will consume.
Third principal component reflects car speed. Unlike for the previous PC, horsepower and curb.weight dominate this component. Negative effect of carb.weight and positive of horsepower. The heavier the car and the less horsepower it has, the slower it is.

Since 3 of 4 of the car's characteristics works in tandem for highering the price of the vehicle, it can explain the spread in question 2.3. For example, cars with equally big carb.size may have different combination of horsepower and engine.size.

```r
round(PC[,1],3)
```

```
## curb.weight engine.size  horsepower highway.mpg
##       0.507       0.500       0.505      -0.488
```

```r
round(PC[,2],3)
```

```
## curb.weight engine.size  horsepower highway.mpg
##      -0.186      -0.636       0.100      -0.742
```

```r
round(PC[,3],3)
```

```
## curb.weight engine.size  horsepower highway.mpg
##      -0.657       0.103       0.726       0.173
```