Graphic Analysis of Lottery Data with Boxplots

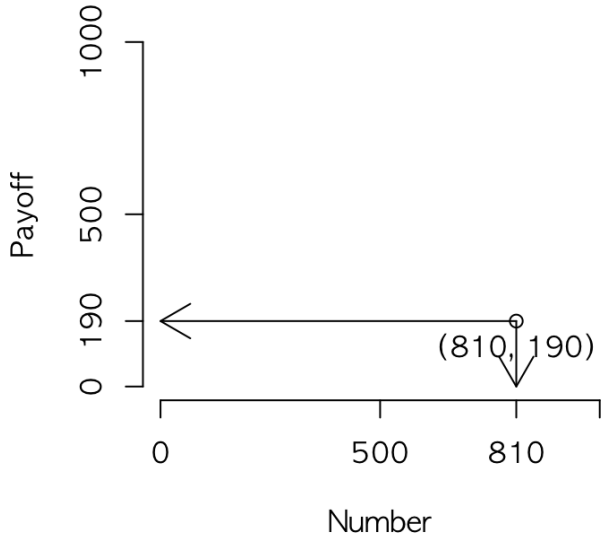
coop711

2015년 3월 14일

Scatter Plot

- 이제 두 변수의 산점도를 그려보자. 산점도는 관찰값의 x좌표와 y좌표를 평면 상에 나타낸 것이다. lottery 자료의 첫날 당첨번호와 당첨금액은 `lottery[1,]` 이므로, 각 좌표를 평면상에 나타내면 text() 함수를 이용하여

```
load("lottery.rda")
attach(lottery)
par(mfrow=c(1,2))
plot(lottery[1,], xlim=c(0,1000), ylim=c(0,1000), axes=F, xlab="Number", yla
b="Payoff")
axis(side=1, at=c(0,500,810,1000), labels=c(0,500,810,1000))
axis(side=2, at=c(0,190, 500 ,1000), labels=c(0,190, 500,1000))
text(lottery[1,], labels=c("(810, 190)"), pos=1)
arrows(x0=810,y0=190,x1=810,y1=0, code=2, length=0.2)
arrows(x0=810,y0=190,x1=0,y1=190, code=2, length=0.2)
```



- 당첨번호 0, 499, 999에 해당하는 당첨금액을 찾아서 평면 상에 나타내면?

```
id.0<-which(lottery$lottery.number==0)
lottery[id.0,]
```

```
##    lottery.number lottery.payoff
## 99              0             96
```

```
id.499<-which(lottery$lottery.number==499)
lottery[id.499,]
```

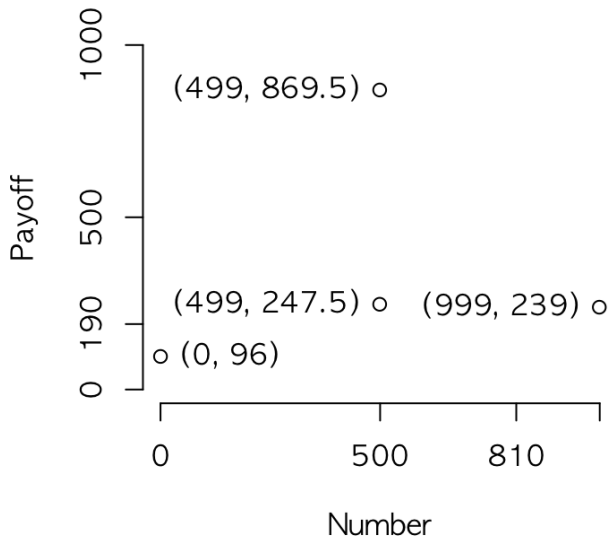
```
##     lottery.number lottery.payoff
## 10             499          869.5
## 132            499          247.5
```

```
id.999<-which(lottery$lottery.number==999)
lottery[id.999,]
```

```
##     lottery.number lottery.payoff
## 168            999            239
```

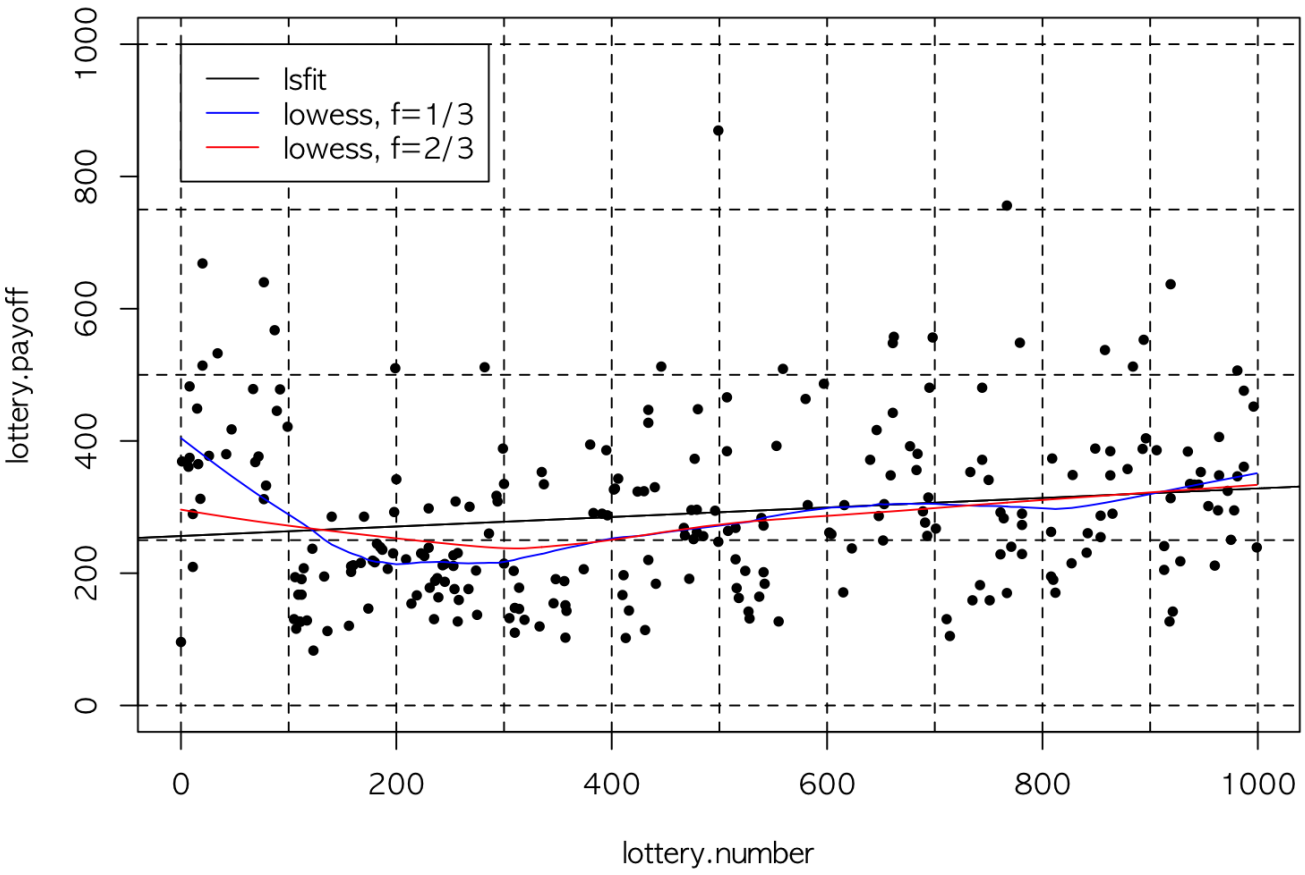
- 파악한 좌표를 평면 상에 points()를 이용하여 표시하고, text() 로 라벨을 붙임.

```
par(mfrow=c(1,2))
plot(lottery[c(id.0,id.499,id.999),], xlim=c(0,1000), ylim=c(0,1000), axes=F, x
lab="Number", ylab="Payoff")
axis(side=1, at=c(0,500,810,1000), labels=c(0,500,810,1000))
axis(side=2, at=c(0,190, 500 ,1000), labels=c(0,190, 500,1000))
text(lottery[id.0,], labels="(0, 96)", pos=4)
text(lottery[c(id.499,id.999),], labels=c("(499, 869.5)", "(499, 247.5)", "(99
9, 239)"), pos=2)
```



- 흐름을 파악하기 위하여 local smoother 를 적용. 최소제곱법으로 구한 1차회귀선과 비교. legend()를 이용하여 범례를 만들 때에도 좌표를 주는 것 이외의 방법을 알아 둘 것.

```
plot(lottery.number, lottery.payoff,pch=20, ylim=c(0,1000))
abline(lsfit(lottery.number, lottery.payoff)$coef)
abline(h=seq(0,1000,by=250),lty=2)
abline(v=seq(0,1000,by=100),lty=2)
abline(lsfit(lottery.number, lottery.payoff)$coef)
lines(lowess(lottery.number,lottery.payoff, f=1/3),col="blue")
lines(lowess(lottery.number,lottery.payoff, f=2/3),col="red")
legend(x=0,y=1000, lty=1, col=c("black","blue","red"), legend=c("lsfit","lowes
s, f=1/3","lowess, f=2/3"))
```



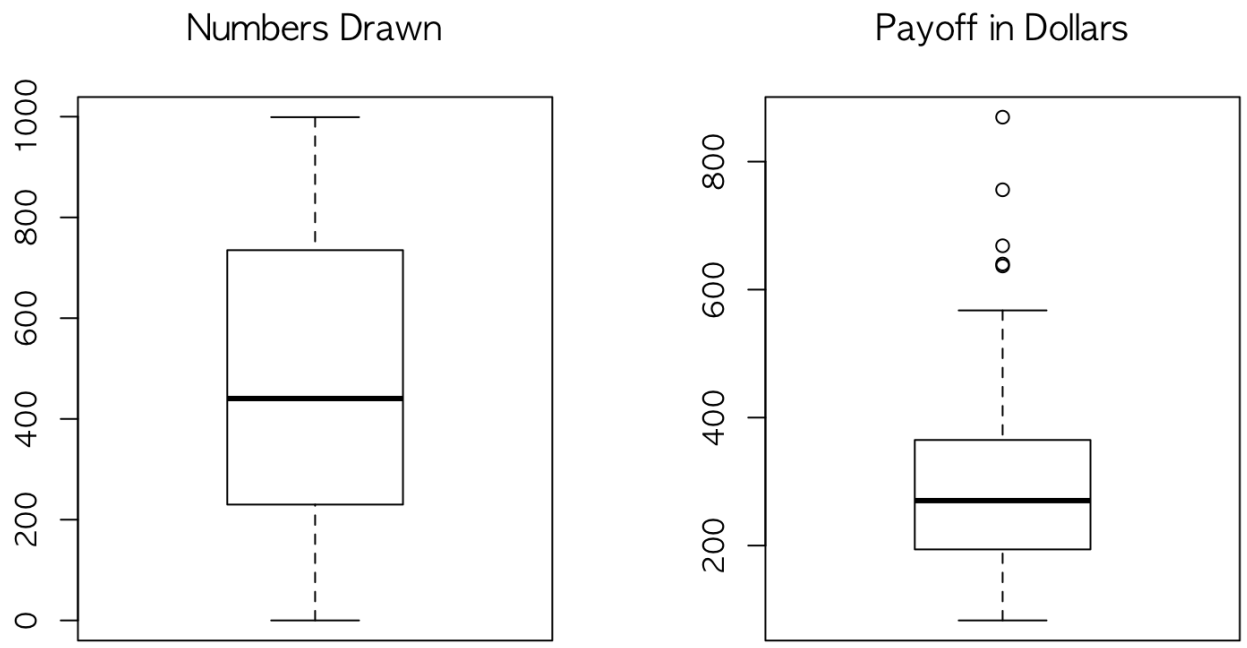
- 이제는 boxplot을 이용하여 자료의 특징을 살펴보자. 단순히 당첨번호와 당첨금액의 boxplot을 그려보는 것은 fivenum()을 확인하는 데 지나지 않으므로 산점도로부터 파악한 사실들을 일깨워 보자. 우선, 다섯 숫자 요약을 살펴보면

```
apply(lottery, 2, fivenum)
```

```
##      lottery.number lottery.payoff
## [1,]            0.0          83.00
## [2,]          230.0         194.00
## [3,]          440.5         270.25
## [4,]          735.0         365.00
## [5,]          999.0         869.50
```

- 당첨번호와 당첨금액의 boxplot을 나란히 그려보면

```
par(mfrow=c(1,2))
boxplot(lottery.number, main="Numbers Drawn")
boxplot(lottery.payoff, main="Payoff in Dollars")
```



- 당첨번호와 당첨금액의 관계를 boxplot()을 이용하여 살펴보려면 먼저 당첨번호를 계급으로 나누어야 함. 이때 cut() 을 이용하여 factor를 생성하게 됨. 먼저 혼동을 없애기 위해 lottery를 lottery.fac에 저장하고, classes.10를 생성. 이때 구간의 모양을 같게 하기 위하여 마지막 값을 어떻게 설정하였는지 유의.

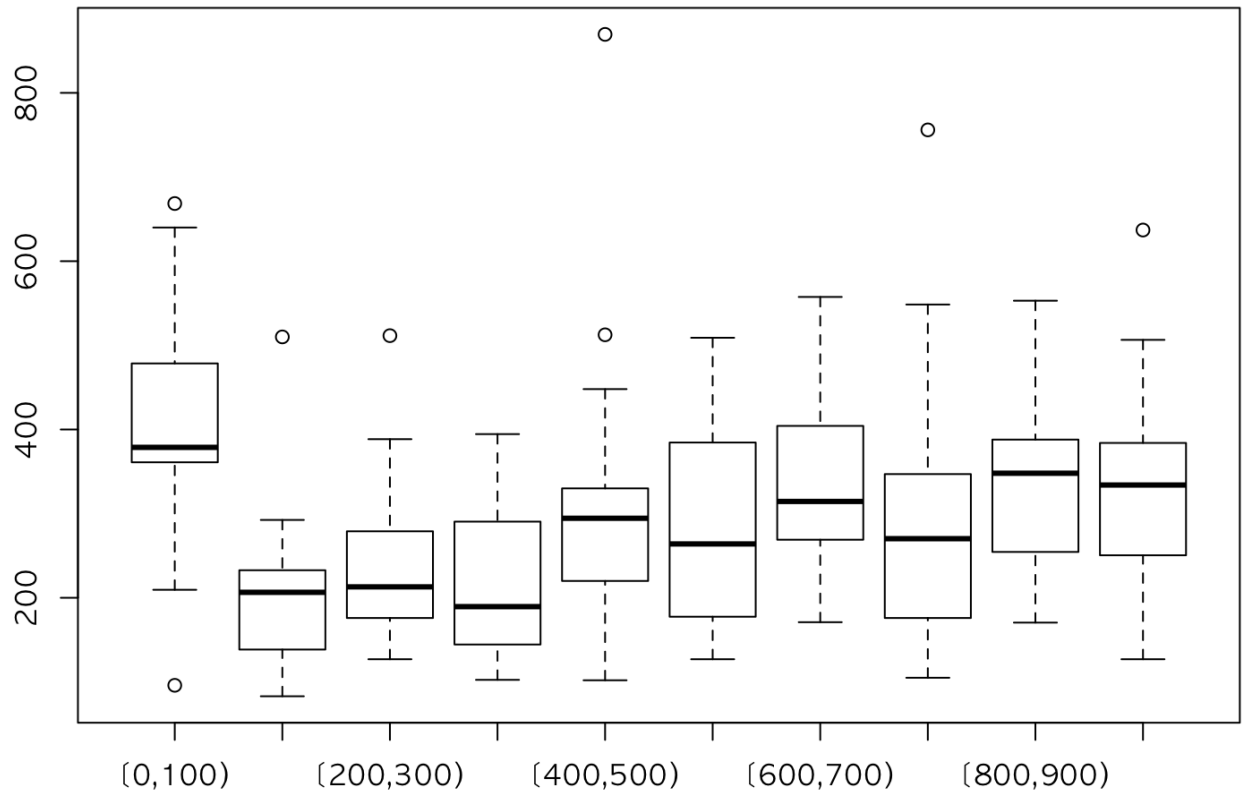
```
lottery.fac<-lottery
lottery.fac$classes.10<-cut(lottery.fac$lottery.number, breaks=c(seq(0,900, b
y=100),999), right=F)
head(lottery.fac)
```

```
##   lottery.number lottery.payoff classes.10
## 1            810          190.0  [800,900)
## 2            156          120.5  [100,200)
## 3            140          285.5  [100,200)
## 4            542          184.0  [500,600)
## 5            507          384.5  [500,600)
## 6            972          324.5  [900,999)
```

```
detach()
attach(lottery.fac)
```

- 이 상태로 당첨번호와 당첨금액의 관계를 boxplot으로 나타내면 관계는 명확히 파악할 수 있으나 x축이 너무 번잡하게 됨.

```
boxplot(lottery.payoff~classes.10, data=lottery.fac)
```

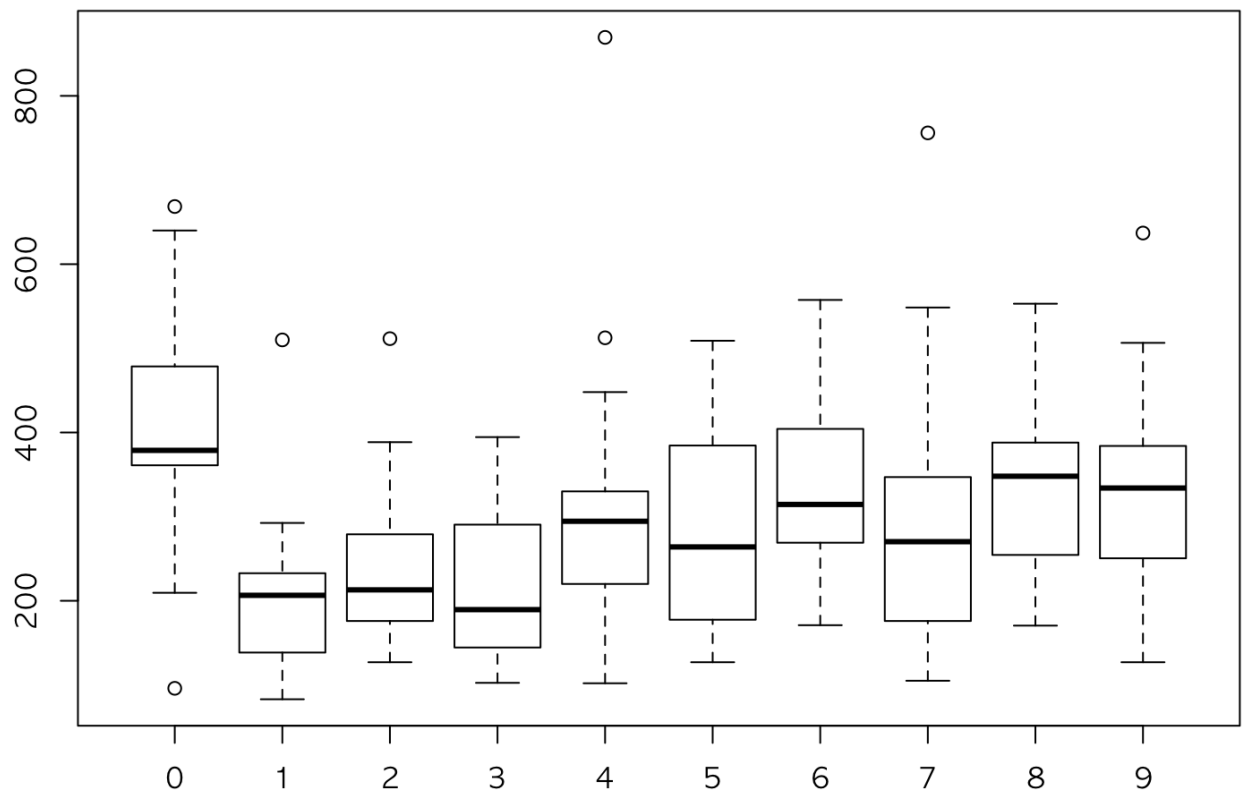


- classes.10 의 labels 들을 구간의 첫 글자로 만들어 주고, 다시 그리면.

```
lottery.fac$classes<-factor(classes.10, labels=0:9)
head(lottery.fac)
```

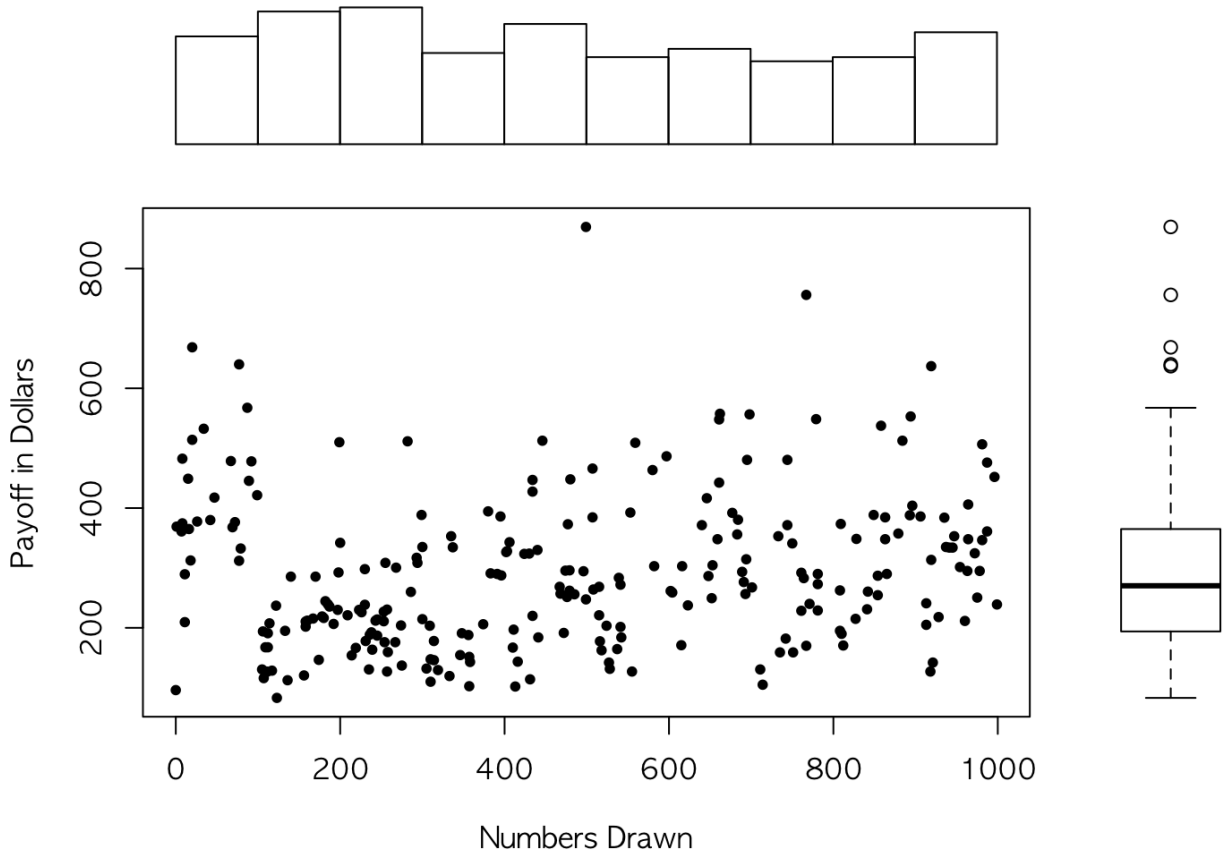
```
##   lottery.number lottery.payoff classes.10 classes
## 1            810          190.0  [800,900)       8
## 2            156          120.5  [100,200)       1
## 3            140          285.5  [100,200)       1
## 4            542          184.0  [500,600)       5
## 5            507          384.5  [500,600)       5
## 6            972          324.5  [900,999)       9
```

```
boxplot(lottery.payoff~classes, data=lottery.fac, main="Payoff by Numbers Draw
n")
```



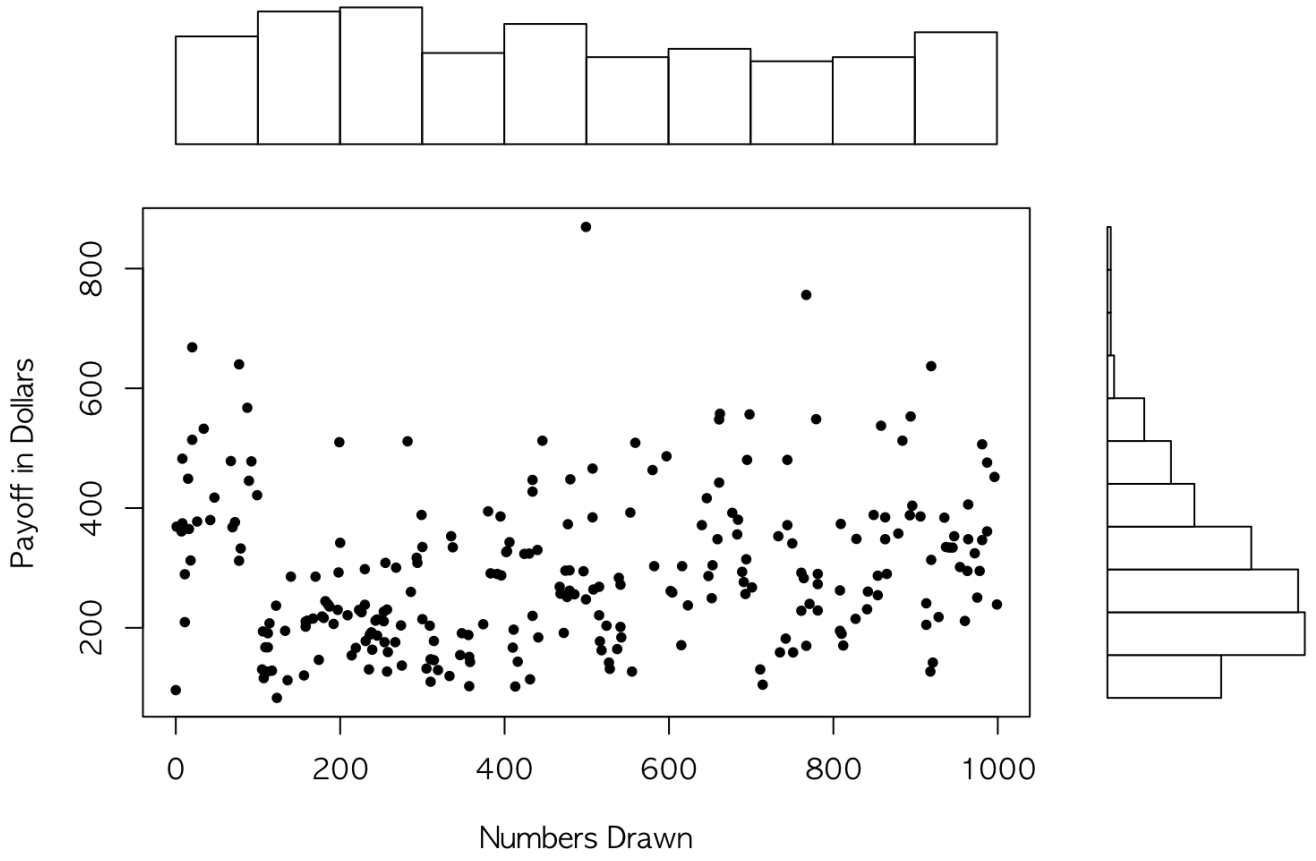
- boxplot() 대신에 그냥 plot()을 하면 어떻게 되는지 시도해 볼 것. 차이점은?
- 이제 산점도에 각 변수의 주변분포(marginal distribution)를 표시해 보자. 당첨번호는 히스토그램으로, 당첨금액은 boxplot 으로 그려 본다.

```
opar<-par(no.readonly=TRUE)
par(fig=c(0,0.8,0,0.8))
plot(lottery.payoff~lottery.number, data=lottery.fac, pch=20, xlab="Numbers Dra
wn", ylab="Payoff in Dollars")
par(fig=c(0,0.8,0.55,1), new=TRUE)
hist(lottery.number, axes=F, ann=F)
par(fig=c(0.65,1,0,0.8), new=TRUE)
boxplot(lottery.payoff, horiz=TRUE, axes=F)
```



- 만일 boxplot 대신에 히스토그램을 눕혀 그리고 싶다면, hist()에는 horiz= 이 없기 때문에 barplot()에서 설정해 주어야 함. 일단, 그려보고 각각이 왜 필요한지 생각해 볼 것.

```
par(fig=c(0,0.8,0,0.8))
plot(lottery.payoff~lottery.number, data=lottery.fac, pch=20, xlab="Numbers Dra
wn", ylab="Payoff in Dollars")
par(fig=c(0,0.8,0.55,1), new=TRUE)
hist(lottery.number, axes=F, ann=F)
par(fig=c(0.7,1,0,0.8), new=TRUE)
barplot(table(cut(lottery.payoff, breaks=11)), horiz=T, space=0, col="white", a
xes=F, axisnames=F)
```



- 이제 당첨금액이 높은 당첨번호들은 숫자가 중복되는 경우가 많고, 당첨번호가 0에서 100 이하인 경우에 당첨금액이 높은지 생각해 보자. detach(lottery)를 하지 않고 deatch()만 해도 되는 이유는 뭘까? save(file=filename, list=ls()) 와 같은 것이 save.image(file=filename) 임. 확인하기를^^

```
detach()
par(opar)
save(file="lottery.RData",list=ls())
```

```
savehistory("lottery.Rhistory")
```