product

ekoh

2020 2 7

# 상관 분석을 통한 지역별 돼지고기 가격 연관성 분석

## 패키지 및 데이터 불러오기

library1 <- c("plyr", "ggplot2", "stringr", "zoo", "corrplot", "RColorBrewer")  
unlist(lapply(library1, require, character.only=TRUE))

## Loading required package: plyr

## Loading required package: ggplot2

## Loading required package: stringr

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: corrplot

## corrplot 0.84 loaded

## Loading required package: RColorBrewer

## [1] TRUE TRUE TRUE TRUE TRUE TRUE

#install.packages("corrplot")  
#install.packages("zoo")  
  
product <- read.csv("product.csv", header = T, fileEncoding = "UTF-8")  
weather <- read.csv("weather.csv", header = T, fileEncoding = "EUC-KR")  
code <- read.csv("code.csv", header = T)  
  
head(product)

## 일자 부류코드 품목코드 지역코드 마트코드 가격  
## 1 2011-01-03 100 111 3511 350401 39600  
## 2 2011-01-03 100 111 3311 330401 40000  
## 3 2011-01-03 100 111 1101 110251 37000  
## 4 2011-01-03 100 111 1101 110401 43900  
## 5 2011-01-03 100 111 1101 110402 39800  
## 6 2011-01-03 100 111 1101 110403 39600

head(weather)

## 지역 기상구분 측정값 일자  
## 1 백령도(기) 강수량 NA 2010-06-02  
## 2 백령도(기) 강수량 3.0 2010-07-02  
## 3 백령도(기) 강수량 NA 2010-08-02  
## 4 백령도(기) 강수량 8.5 2010-09-02  
## 5 백령도(기) 강수량 37.5 2010-10-02  
## 6 백령도(기) 강수량 NA 2010-11-02

head(code)

## 구분코드 구분코드설명 분류코드 분류코드설명  
## 1 1100 부류코드 100 식량작물  
## 2 1100 부류코드 200 채소류  
## 3 1100 부류코드 300 측용작물  
## 4 1100 부류코드 400 과일류  
## 5 1100 부류코드 500 축산물  
## 6 1200 품목코드 111 쌀

## 데이터 가공

colnames(product) <- c('date', 'category', 'item', 'region', 'mart', 'price')  
  
category <- subset(code, code$"구분코드설명" =="품목코드")  
category

## 구분코드 구분코드설명 분류코드 분류코드설명  
## 6 1200 품목코드 111 쌀  
## 7 1200 품목코드 211 배추  
## 8 1200 품목코드 214 상추  
## 9 1200 품목코드 224 호박  
## 10 1200 품목코드 245 양파  
## 11 1200 품목코드 256 파프리카  
## 12 1200 품목코드 312 참깨   
## 13 1200 품목코드 411 사과  
## 14 1200 품목코드 514 돼지고기  
## 15 1200 품목코드 515 닭고기

colnames(category) <- c('code', 'exp', 'item', 'name')  
  
head(product)

## date category item region mart price  
## 1 2011-01-03 100 111 3511 350401 39600  
## 2 2011-01-03 100 111 3311 330401 40000  
## 3 2011-01-03 100 111 1101 110251 37000  
## 4 2011-01-03 100 111 1101 110401 43900  
## 5 2011-01-03 100 111 1101 110402 39800  
## 6 2011-01-03 100 111 1101 110403 39600

category

## code exp item name  
## 6 1200 품목코드 111 쌀  
## 7 1200 품목코드 211 배추  
## 8 1200 품목코드 214 상추  
## 9 1200 품목코드 224 호박  
## 10 1200 품목코드 245 양파  
## 11 1200 품목코드 256 파프리카  
## 12 1200 품목코드 312 참깨   
## 13 1200 품목코드 411 사과  
## 14 1200 품목코드 514 돼지고기  
## 15 1200 품목코드 515 닭고기

total.pig <- product[which(product$item == 514),]  
head(total.pig, 10)

## date category item region mart price  
## 211 2011-01-03 500 514 2200 220063 1500  
## 212 2011-01-03 500 514 2200 220401 1380  
## 213 2011-01-03 500 514 3911 390401 1980  
## 214 2011-01-03 500 514 3511 350401 1380  
## 215 2011-01-03 500 514 3311 330401 1890  
## 216 2011-01-03 500 514 3211 320401 1280  
## 217 2011-01-03 500 514 3145 310403 1280  
## 218 2011-01-03 500 514 3111 310402 2050  
## 219 2011-01-03 500 514 2601 260401 1980  
## 220 2011-01-03 500 514 2501 250403 1380

region <- subset(code, code$"구분코드설명"=='지역코드')  
region

## 구분코드 구분코드설명 분류코드 분류코드설명  
## 16 1300 지역코드 1101 서울   
## 17 1300 지역코드 2100 부산  
## 18 1300 지역코드 2200 대구  
## 19 1300 지역코드 2300 인천  
## 20 1300 지역코드 2401 광주  
## 21 1300 지역코드 2501 대전  
## 22 1300 지역코드 2601 울산  
## 23 1300 지역코드 3111 수원  
## 24 1300 지역코드 3113 의정부  
## 25 1300 지역코드 3145 용인  
## 26 1300 지역코드 3211 춘천  
## 27 1300 지역코드 3311 청주  
## 28 1300 지역코드 3511 전주  
## 29 1300 지역코드 3613 순천  
## 30 1300 지역코드 3711 포항  
## 31 1300 지역코드 3714 안동  
## 32 1300 지역코드 3814 창원  
## 33 1300 지역코드 3911 제주

colnames(region) <- c('code', 'exp', 'region', 'name')  
region

## code exp region name  
## 16 1300 지역코드 1101 서울   
## 17 1300 지역코드 2100 부산  
## 18 1300 지역코드 2200 대구  
## 19 1300 지역코드 2300 인천  
## 20 1300 지역코드 2401 광주  
## 21 1300 지역코드 2501 대전  
## 22 1300 지역코드 2601 울산  
## 23 1300 지역코드 3111 수원  
## 24 1300 지역코드 3113 의정부  
## 25 1300 지역코드 3145 용인  
## 26 1300 지역코드 3211 춘천  
## 27 1300 지역코드 3311 청주  
## 28 1300 지역코드 3511 전주  
## 29 1300 지역코드 3613 순천  
## 30 1300 지역코드 3711 포항  
## 31 1300 지역코드 3714 안동  
## 32 1300 지역코드 3814 창원  
## 33 1300 지역코드 3911 제주

head(total.pig)

## date category item region mart price  
## 211 2011-01-03 500 514 2200 220063 1500  
## 212 2011-01-03 500 514 2200 220401 1380  
## 213 2011-01-03 500 514 3911 390401 1980  
## 214 2011-01-03 500 514 3511 350401 1380  
## 215 2011-01-03 500 514 3311 330401 1890  
## 216 2011-01-03 500 514 3211 320401 1280

day.pig <- merge(total.pig, region, by = 'region', all = T)  
head(day.pig, 10)

## region date category item mart price code exp name  
## 1 1101 2012-07-18 500 514 110251 1800 1300 지역코드 서울   
## 2 1101 2012-06-19 500 514 110402 1580 1300 지역코드 서울   
## 3 1101 2012-07-05 500 514 110406 1780 1300 지역코드 서울   
## 4 1101 2012-04-30 500 514 110407 1490 1300 지역코드 서울   
## 5 1101 2011-09-05 500 514 110403 1980 1300 지역코드 서울   
## 6 1101 2011-10-31 500 514 110251 1800 1300 지역코드 서울   
## 7 1101 2012-04-30 500 514 110403 1750 1300 지역코드 서울   
## 8 1101 2011-01-21 500 514 110406 2340 1300 지역코드 서울   
## 9 1101 2012-03-26 500 514 110402 1750 1300 지역코드 서울   
## 10 1101 2012-06-21 500 514 110406 1780 1300 지역코드 서울

### day.pig 데이터를 일별로 정렬한 후, 지역별로 돼지고기의 평균가격을 구하여 생성한 데이터프레임을 지역별 이름으로 나누어 total.pig.mean이라는 리스트 형태의 데이터 생성

#### ddply() 이해를 위한 예제

x <- data.frame(  
 Date=as.Date(c('2013-10-01', '2013-10-02', '2013-10-02', '2013-10-02', '2013-10-01', '2013-10-02', '2013-10-02')),  
 Category=factor(c('First', 'First', 'First', 'Second', 'Third', 'Third', 'Second')),  
 Frequency=c(10,15,5,2,14,20,3))  
  
head(x)

## Date Category Frequency  
## 1 2013-10-01 First 10  
## 2 2013-10-02 First 15  
## 3 2013-10-02 First 5  
## 4 2013-10-02 Second 2  
## 5 2013-10-01 Third 14  
## 6 2013-10-02 Third 20

ddply(x, .(Date, Category), summarize, Sum\_F=sum(Frequency))

## Date Category Sum\_F  
## 1 2013-10-01 First 10  
## 2 2013-10-01 Third 14  
## 3 2013-10-02 First 20  
## 4 2013-10-02 Second 5  
## 5 2013-10-02 Third 20

# dlply() 이해를 위한 예제  
x <- data.frame(  
 Date=as.Date(c('2013-10-01', '2013-10-02', '2013-10-02', '2013-10-02', '2013-10-01', '2013-10-02', '2013-10-02')),  
 Category=factor(c('First', 'First', 'First', 'Second', 'Third', 'Third', 'Second')),  
 Frequency=c(10,15,5,2,14,20,3))  
  
dlply(x, .(Date), summarize, Sum\_F=sum(Frequency))

## $`2013-10-01`  
## Sum\_F  
## 1 24  
##   
## $`2013-10-02`  
## Sum\_F  
## 1 45  
##   
## attr(,"split\_type")  
## [1] "data.frame"  
## attr(,"split\_labels")  
## Date  
## 1 2013-10-01  
## 2 2013-10-02

#### 적용

head(day.pig)

## region date category item mart price code exp name  
## 1 1101 2012-07-18 500 514 110251 1800 1300 지역코드 서울   
## 2 1101 2012-06-19 500 514 110402 1580 1300 지역코드 서울   
## 3 1101 2012-07-05 500 514 110406 1780 1300 지역코드 서울   
## 4 1101 2012-04-30 500 514 110407 1490 1300 지역코드 서울   
## 5 1101 2011-09-05 500 514 110403 1980 1300 지역코드 서울   
## 6 1101 2011-10-31 500 514 110251 1800 1300 지역코드 서울

p1 <- ddply(day.pig, .(date), summarise, name=name, region=region, price=price)  
p2 <- ddply(p1, .(date, name), summarise, mean.price = mean(price))  
head(p2)

## date name mean.price  
## 1 2011-01-03 광주 1610.000  
## 2 2011-01-03 대구 1627.500  
## 3 2011-01-03 대전 1335.000  
## 4 2011-01-03 부산 1937.500  
## 5 2011-01-03 서울 1738.333  
## 6 2011-01-03 수원 2050.000

total.pig.mean <- dlply(p2, .(name))  
# total.pig.mean <- dlply(ddply(ddply(day.pig, .(date), summarise, name=name, region=region, price=price), .(date, name), summarise, mean.price=mean(price)), .(name))

### 각 지역별 데이터의 크기 확인하기

for (i in 1:length(total.pig.mean)) {  
 cat(names(total.pig.mean)[i],"의 데이터의 길이는",   
 nrow(total.pig.mean[[i]]),"이다\n")  
}

## 광주 의 데이터의 길이는 745 이다  
## 대구 의 데이터의 길이는 745 이다  
## 대전 의 데이터의 길이는 745 이다  
## 부산 의 데이터의 길이는 745 이다  
## 서울 의 데이터의 길이는 745 이다  
## 수원 의 데이터의 길이는 745 이다  
## 순천 의 데이터의 길이는 477 이다  
## 안동 의 데이터의 길이는 468 이다  
## 용인 의 데이터의 길이는 268 이다  
## 울산 의 데이터의 길이는 745 이다  
## 의정부 의 데이터의 길이는 477 이다  
## 인천 의 데이터의 길이는 745 이다  
## 전주 의 데이터의 길이는 745 이다  
## 제주 의 데이터의 길이는 745 이다  
## 창원 의 데이터의 길이는 477 이다  
## 청주 의 데이터의 길이는 745 이다  
## 춘천 의 데이터의 길이는 743 이다  
## 포항 의 데이터의 길이는 476 이다

head(day.pig)

## region date category item mart price code exp name  
## 1 1101 2012-07-18 500 514 110251 1800 1300 지역코드 서울   
## 2 1101 2012-06-19 500 514 110402 1580 1300 지역코드 서울   
## 3 1101 2012-07-05 500 514 110406 1780 1300 지역코드 서울   
## 4 1101 2012-04-30 500 514 110407 1490 1300 지역코드 서울   
## 5 1101 2011-09-05 500 514 110403 1980 1300 지역코드 서울   
## 6 1101 2011-10-31 500 514 110251 1800 1300 지역코드 서울

day.pig <- day.pig[!day.pig$name %in% c("의정부", "용인", "창원", "안동", "포항", "순천", "춘천"),]

### day.pig 데이터를 지역(region), 일자(date)별로 돼지고기 평균가격 구하여 pig.region.daily.mean 데이터를 생성

pig.region.daily.mean <- ddply(day.pig, .(name, region, date), summarise, mean.price = mean(price))  
head(pig.region.daily.mean, n=10)

## name region date mean.price  
## 1 광주 2401 2011-01-03 1610  
## 2 광주 2401 2011-01-04 1610  
## 3 광주 2401 2011-01-05 1610  
## 4 광주 2401 2011-01-06 1390  
## 5 광주 2401 2011-01-07 1390  
## 6 광주 2401 2011-01-10 1390  
## 7 광주 2401 2011-01-11 1390  
## 8 광주 2401 2011-01-12 1390  
## 9 광주 2401 2011-01-13 1635  
## 10 광주 2401 2011-01-14 1735

pig.region.monthly.mean <- ddply(pig.region.daily.mean,  
 .(name, region, month=str\_sub(pig.region.daily.mean$date,1,7)),  
 summarise, mean.price=mean(mean.price))  
head(pig.region.monthly.mean, 10)

## name region month mean.price  
## 1 광주 2401 2011-01 1782.619  
## 2 광주 2401 2011-02 2073.235  
## 3 광주 2401 2011-03 1771.364  
## 4 광주 2401 2011-04 1785.000  
## 5 광주 2401 2011-05 1956.750  
## 6 광주 2401 2011-06 2357.857  
## 7 광주 2401 2011-07 2452.976  
## 8 광주 2401 2011-08 2253.182  
## 9 광주 2401 2011-09 1997.375  
## 10 광주 2401 2011-10 1731.375

pig.region.yearly.mean <- ddply(pig.region.daily.mean,  
 .(name, region, year=str\_sub(pig.region.daily.mean$date,1,4)),  
 summarise, mean.price=mean(mean.price))  
head(pig.region.yearly.mean, 10)

## name region year mean.price  
## 1 광주 2401 2011 1990.312  
## 2 광주 2401 2012 1658.922  
## 3 광주 2401 2013 1573.522  
## 4 대구 2200 2011 2031.474  
## 5 대구 2200 2012 1714.234  
## 6 대구 2200 2013 1533.105  
## 7 대전 2501 2011 1969.667  
## 8 대전 2501 2012 1691.904  
## 9 대전 2501 2013 1527.339  
## 10 부산 2100 2011 2185.958

## 상관분석 및 데이터 시각화

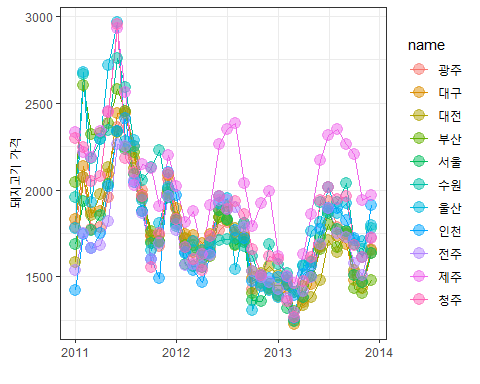
### 월별 돼지고기 가격 시각화

#### geom\_line

pig.region.monthly.mean$month <- as.Date(as.yearmon(pig.region.monthly.mean$month, "%Y-%m"))  
head(pig.region.monthly.mean)

## name region month mean.price  
## 1 광주 2401 2011-01-01 1782.619  
## 2 광주 2401 2011-02-01 2073.235  
## 3 광주 2401 2011-03-01 1771.364  
## 4 광주 2401 2011-04-01 1785.000  
## 5 광주 2401 2011-05-01 1956.750  
## 6 광주 2401 2011-06-01 2357.857

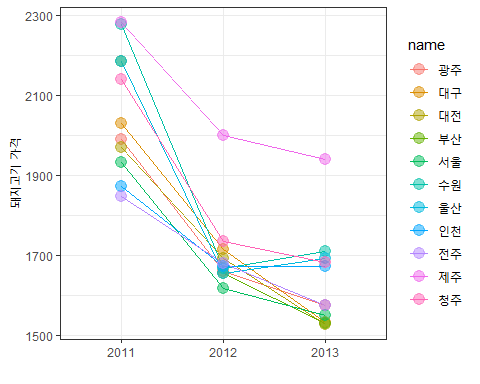
ggplot(pig.region.monthly.mean, aes(x=month, y=mean.price, colour=name, group=name)) +  
 geom\_line() + theme\_bw() + geom\_point(size=6, shape=20,alpha=0.5) +  
 ylab("돼지고기 가격") + xlab("")



### 지역별 연간 돼지고기 평균가격 시각화

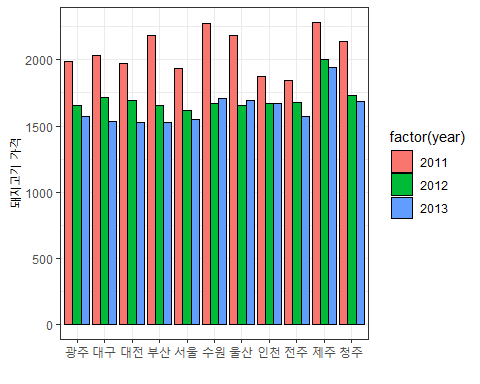
#### geom\_line

ggplot(pig.region.yearly.mean, aes(x=year, y=mean.price, colour=name, group=name)) +  
 geom\_line() + geom\_point(size=6,shape=20, alpha=0.5) + theme\_bw() +  
 ylab("돼지고기 가격") + xlab("")



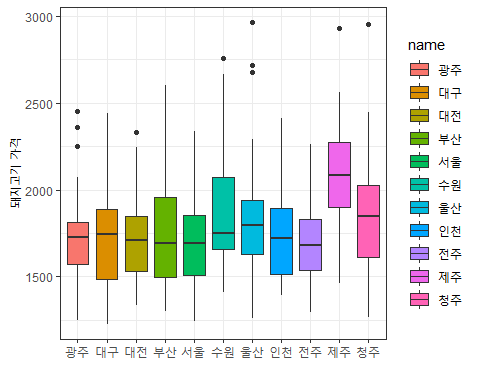
#### geom\_bar

ggplot(pig.region.yearly.mean, aes(x=name, y=mean.price, fill=factor(year))) +  
 theme\_bw() +  
 geom\_bar(stat="identity", position="dodge", colour="black") +  
 ylab("돼지고기 가격") + xlab("")



#### geom\_boxplot

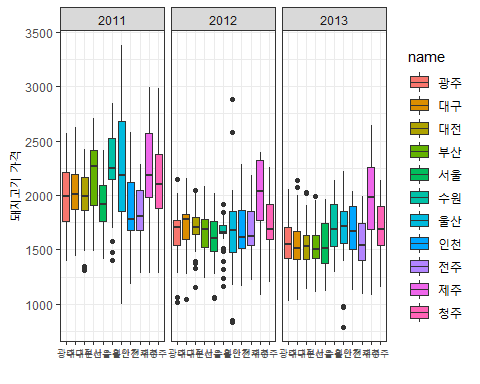
ggplot(pig.region.monthly.mean, aes(x=name, y=mean.price, fill=name)) +  
 theme\_bw() +  
 geom\_boxplot() +   
 ylab("돼지고기 가격") +  
 xlab("")



year.pig <- data.frame(name = pig.region.daily.mean$name,  
 region = pig.region.daily.mean$region,  
 date=pig.region.daily.mean$date,  
 year=str\_sub(pig.region.daily.mean$date,1,4),  
 mean.price=pig.region.daily.mean$mean.price)  
head(year.pig)

## name region date year mean.price  
## 1 광주 2401 2011-01-03 2011 1610  
## 2 광주 2401 2011-01-04 2011 1610  
## 3 광주 2401 2011-01-05 2011 1610  
## 4 광주 2401 2011-01-06 2011 1390  
## 5 광주 2401 2011-01-07 2011 1390  
## 6 광주 2401 2011-01-10 2011 1390

ggplot(year.pig, aes(x=name, y=mean.price, fill=name)) +  
 geom\_boxplot() +  
 theme\_bw() +  
 facet\_wrap(~year, scales = 'fixed') +  
 xlab('') + ylab("돼지고기 가격") +  
 theme(axis.text.x = element\_text(size=7))

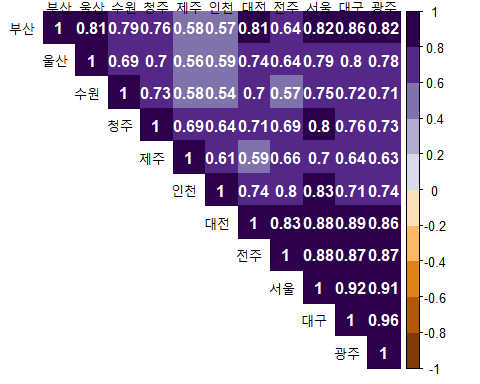


### 상관관계가 높은 도시들을 묶어 가격변화 살펴보기

temp <- dlply(pig.region.daily.mean, .(name), summarise, mean.price)  
pig.region <- data.frame("서울"=unlist(temp$"서울"),  
 "부산"=unlist(temp$"부산"), "대구"=unlist(temp$"대구"), "인천"=unlist(temp$"인천"), "광주"=unlist(temp$"광주"), "대전"=unlist(temp$"대전"), "울산"=unlist(temp$"울산"), "수원"=unlist(temp$"수원"), "청주"=unlist(temp$"청주"), "전주"=unlist(temp$"전주"), "제주"=unlist(temp$"제주"))  
head(pig.region)

## 서울 부산 대구 인천 광주 대전 울산 수원 청주 전주 제주  
## ..11 1738.333 1937.5 1627.5 1280 1610 1335 1980 2050 1890 1380 1980  
## ..12 1738.333 1937.5 1627.5 1280 1610 1560 1980 2050 1890 1380 1980  
## ..13 1738.333 1937.5 1627.5 1280 1610 1560 1980 2050 2010 1380 1980  
## ..14 1538.333 1692.5 1442.5 1380 1390 1310 1000 2050 2010 1380 1980  
## ..15 1538.333 1692.5 1492.5 1380 1390 1310 1000 2050 2010 1380 2140  
## ..16 1543.333 1692.5 1492.5 1380 1390 1310 1000 2110 2340 1380 2140

cor\_pig <- cor(pig.region)  
  
corrplot(cor\_pig, method="color", type="upper", order="hclust", addCoef.col = "white", tl.srt = 0, tl.col = "black", tl.cex=1, col=brewer.pal(n=10, name="PuOr"))



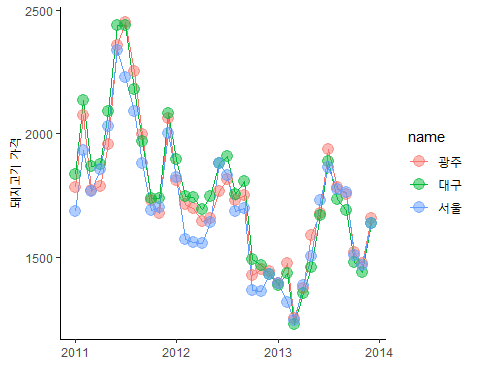
#### 월별 데이터 가공

pig.region.monthly.mean$month <- as.Date(as.yearmon(pig.region.monthly.mean$month, "%Y-%m"))  
head(pig.region.monthly.mean)

## name region month mean.price  
## 1 광주 2401 2011-01-01 1782.619  
## 2 광주 2401 2011-02-01 2073.235  
## 3 광주 2401 2011-03-01 1771.364  
## 4 광주 2401 2011-04-01 1785.000  
## 5 광주 2401 2011-05-01 1956.750  
## 6 광주 2401 2011-06-01 2357.857

#### 광주, 대구, 서울 지역의 2011~2013년 월별 돼지고기 가격 시계열 그래프

ggplot(pig.region.monthly.mean[pig.region.monthly.mean$name %in% c("광주", "대구", "서울 "),],  
 aes(x=month, y=mean.price, colour=name, group=name)) +  
 geom\_line() + theme\_classic() +   
 geom\_point(size = 6, shape=20, alpha=0.5) +  
 ylab("돼지고기 가격") + xlab("")



## 데이터 저장하기

write.csv(pig.region, "pig.region.csv", fileEncoding = "UTF-8")  
write.csv(pig.region.monthly.mean, "pig.region.monthly.mean.csv", fileEncoding = "UTF-8")

# 공적분 검정을 통한 농축산물 소매가격 연관성 분석

#install.packages("stringr")  
#install.packages("urca")  
#install.packages("gridExtra")  
  
library2 <- c("plyr", "stringr", "urca", "ggplot2", "zoo", "gridExtra")  
unlist(lapply(library2, require, character.only = TRUE))

## Loading required package: urca

## Loading required package: gridExtra

## [1] TRUE TRUE TRUE TRUE TRUE TRUE

## 데이터 불러오기

# 위에서 불러온 파일로 사용

## 데이터 가공하기

str(product)

## 'data.frame': 253418 obs. of 6 variables:  
## $ date : Factor w/ 745 levels "2011-01-03","2011-01-04",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ category: int 100 100 100 100 100 100 100 100 100 100 ...  
## $ item : int 111 111 111 111 111 111 111 111 111 111 ...  
## $ region : int 3511 3311 1101 1101 1101 1101 1101 1101 2100 2100 ...  
## $ mart : int 350401 330401 110251 110401 110402 110403 110405 110406 210022 210401 ...  
## $ price : int 39600 40000 37000 43900 39800 39600 42000 43800 42000 36800 ...

head(product, 10)

## date category item region mart price  
## 1 2011-01-03 100 111 3511 350401 39600  
## 2 2011-01-03 100 111 3311 330401 40000  
## 3 2011-01-03 100 111 1101 110251 37000  
## 4 2011-01-03 100 111 1101 110401 43900  
## 5 2011-01-03 100 111 1101 110402 39800  
## 6 2011-01-03 100 111 1101 110403 39600  
## 7 2011-01-03 100 111 1101 110405 42000  
## 8 2011-01-03 100 111 1101 110406 43800  
## 9 2011-01-03 100 111 2100 210022 42000  
## 10 2011-01-03 100 111 2100 210401 36800

temp <- ddply(product, .(item, date), summarise, mean.price = mean(price))  
head(temp,10)

## item date mean.price  
## 1 111 2011-01-03 40953.33  
## 2 111 2011-01-04 40993.33  
## 3 111 2011-01-05 41153.33  
## 4 111 2011-01-06 41020.00  
## 5 111 2011-01-07 41013.33  
## 6 111 2011-01-10 41180.00  
## 7 111 2011-01-11 41080.00  
## 8 111 2011-01-12 41146.67  
## 9 111 2011-01-13 41146.67  
## 10 111 2011-01-14 41146.67

### 일별 농축산물의 평균 소매가격 추출하기

category <- subset(code, code$"구분코드설명"=="품목코드")  
category

## 구분코드 구분코드설명 분류코드 분류코드설명  
## 6 1200 품목코드 111 쌀  
## 7 1200 품목코드 211 배추  
## 8 1200 품목코드 214 상추  
## 9 1200 품목코드 224 호박  
## 10 1200 품목코드 245 양파  
## 11 1200 품목코드 256 파프리카  
## 12 1200 품목코드 312 참깨   
## 13 1200 품목코드 411 사과  
## 14 1200 품목코드 514 돼지고기  
## 15 1200 품목코드 515 닭고기

colnames(category) <- c('code', 'exp', 'item', 'name')

date.item.mean <- merge(temp, category, by = "item")  
head(date.item.mean)

## item date mean.price code exp name  
## 1 111 2011-01-03 40953.33 1200 품목코드 쌀  
## 2 111 2011-01-04 40993.33 1200 품목코드 쌀  
## 3 111 2011-01-05 41153.33 1200 품목코드 쌀  
## 4 111 2011-01-06 41020.00 1200 품목코드 쌀  
## 5 111 2011-01-07 41013.33 1200 품목코드 쌀  
## 6 111 2011-01-10 41180.00 1200 품목코드 쌀

### 월별 데이터를 필요한 변수들만 구성하여 생성

month.item.mean <- ddply(date.item.mean, .(name, item, month = str\_sub(as.character.Date(date), 1, 7)),  
summarise, mean.price=mean(mean.price))  
  
head(month.item.mean, 10)

## name item month mean.price  
## 1 닭고기 515 2011-01 5517.809  
## 2 닭고기 515 2011-02 6612.114  
## 3 닭고기 515 2011-03 6983.201  
## 4 닭고기 515 2011-04 6911.101  
## 5 닭고기 515 2011-05 6004.349  
## 6 닭고기 515 2011-06 5242.317  
## 7 닭고기 515 2011-07 5971.546  
## 8 닭고기 515 2011-08 6314.089  
## 9 닭고기 515 2011-09 5890.317  
## 10 닭고기 515 2011-10 5751.018

temp <- dlply(date.item.mean, .(name), summarise, mean.price)

### 일간 품목별 평균 데이터

daily.product <- data.frame("쌀"=unlist(temp$"쌀"), "배추"=unlist(temp$"배추"),  
"상추"=unlist(temp$"상추"), "호박"=unlist(temp$"호박"), "양파"=unlist(temp$"양파"), "파프리카"=unlist(temp$"파프리카"), "참깨"=unlist(temp$"참깨"), "사과"=unlist(temp$"사과"), "돼지고기"=unlist(temp$"돼지고기"), "닭고기"=unlist(temp$"닭고기"))  
head(daily.product, 10)

## 쌀 배추 상추 호박 양파 파프리카 참깨 사과  
## ..11 40953.33 4251.818 642.9024 1123.889 1903.333 1015.2963 12133.33 24904.00  
## ..12 40993.33 4168.182 633.3095 1134.259 1877.667 1015.1071 12133.33 24904.00  
## ..13 41153.33 4375.833 629.5227 1145.370 1864.333 1015.1071 12133.33 25176.73  
## ..14 41020.00 4609.167 651.7826 1231.321 2070.000 1043.1071 12133.33 25358.55  
## ..15 41013.33 4609.167 653.9565 1227.925 2071.667 1043.1071 12133.33 25994.91  
## ..16 41180.00 4399.167 648.2391 1275.849 2075.000 1022.7500 12133.33 26176.73  
## ..17 41080.00 4422.500 642.3696 1277.222 2075.000 1025.2857 12133.33 27027.64  
## ..18 41146.67 4495.000 627.4783 1303.774 2067.667 1017.8621 12133.33 27482.18  
## ..19 41146.67 4521.111 635.1522 1350.185 1856.333 1006.8276 12040.00 27320.09  
## ..110 41146.67 4521.111 638.8478 1365.472 1843.000 997.5172 12040.00 27320.09  
## 돼지고기 닭고기  
## ..11 1653.000 5262.524  
## ..12 1683.000 5228.773  
## ..13 1687.000 5263.318  
## ..14 1504.333 5289.696  
## ..15 1516.333 5130.136  
## ..16 1530.333 5201.000  
## ..17 1512.667 5306.045  
## ..18 1512.667 5306.045  
## ..19 1752.000 5656.955  
## ..110 1748.000 5647.087

### 월간 품목별 평균 데이터

temp <- dlply(month.item.mean, .(name), summarise, mean.price)  
monthly.product <- data.frame("쌀"=unlist(temp$"쌀"), "배추"=unlist(temp$"배추"), "상추"=unlist(temp$"상추"), "호박"=unlist(temp$"호박"), "양파"=unlist(temp$"양파"), "파프리카"=unlist(temp$"파프리카"), "참깨"=unlist(temp$"참깨"), "사과"=unlist(temp$"사과"), "돼지고기"=unlist(temp$"돼지고기"), "닭고기"=unlist(temp$"닭고기"))  
head(monthly.product)

## 쌀 배추 상추 호박 양파 파프리카 참깨 사과  
## ..11 41286.35 4728.411 640.2479 1444.737 2000.267 1025.0198 12075.56 27870.33  
## ..12 41957.45 4804.518 641.9159 1544.894 2254.595 1079.0852 11934.12 28694.18  
## ..13 42763.86 4683.778 616.6619 1461.145 2180.592 1008.6646 11814.55 27635.08  
## ..14 44331.37 3301.397 589.1966 1090.387 1635.318 988.8949 11812.00 26181.89  
## ..15 44702.17 1430.533 666.4199 1060.357 1313.533 885.4103 11812.00 26041.14  
## ..16 45076.56 1216.767 631.5808 1196.714 1190.547 849.5205 11812.00 26796.51  
## 돼지고기 닭고기  
## ..11 1780.313 5517.809  
## ..12 2131.667 6612.114  
## ..13 1899.864 6983.201  
## ..14 1958.794 6911.101  
## ..15 2138.150 6004.349  
## ..16 2459.933 5242.317

## 공적분 검정

for (i in 1:9) {  
 for (j in 1:9) {  
 if ((i+j) < 11) {  
 jc <- ca.jo(data.frame(daily.product[,i], daily.product[,i+j]), type = "trace", K=2, ecdet="const")  
 if (jc@teststat[1] > jc@cval[1]) {  
 if(jc@V[1,1]\*jc@V[2,1]>0){  
 cat(colnames(monthly.product)[i],"와",colnames(monthly.product)[i+j],": 음의 공적분 관계가 있다.", "\n")  
 } else {  
 cat(colnames(monthly.product)[i],"와",colnames(monthly.product)[i+j],": 양의 공적분 관계가 있다.", "\n")  
 }  
 }}  
 }  
}

## 상추 와 호박 : 양의 공적분 관계가 있다.   
## 상추 와 사과 : 음의 공적분 관계가 있다.   
## 상추 와 돼지고기 : 양의 공적분 관계가 있다.   
## 상추 와 닭고기 : 음의 공적분 관계가 있다.   
## 호박 와 닭고기 : 음의 공적분 관계가 있다.   
## 사과 와 닭고기 : 음의 공적분 관계가 있다.

### 상추와 호박의 공적분 분석 결과 살펴보기

output <- ca.jo(data.frame(daily.product[,3], daily.product[,4]), type="trace", K=2, ecdet="const")  
summary(output)

##   
## ######################   
## # Johansen-Procedure #   
## ######################   
##   
## Test type: trace statistic , without linear trend and constant in cointegration   
##   
## Eigenvalues (lambda):  
## [1] 1.576451e-02 1.090154e-02 3.469447e-18  
##   
## Values of teststatistic and critical values of test:  
##   
## test 10pct 5pct 1pct  
## r <= 1 | 8.14 7.52 9.24 12.97  
## r = 0 | 19.95 17.85 19.96 24.60  
##   
## Eigenvectors, normalised to first column:  
## (These are the cointegration relations)  
##   
## daily.product...3..l2 daily.product...4..l2 constant  
## daily.product...3..l2 1.00000 1.00000000 1.00000  
## daily.product...4..l2 -1.05189 0.06474024 -25.44892  
## constant 673.25720 -980.34611931 -278264.55826  
##   
## Weights W:  
## (This is the loading matrix)  
##   
## daily.product...3..l2 daily.product...4..l2 constant  
## daily.product...3..d -0.003869091 -0.01182244 4.097721e-21  
## daily.product...4..d 0.019079144 -0.01172547 6.898609e-22

#### 필요한 분석결과 추출

output@teststat[1]

## [1] 8.144317

output@cval[1]

## [1] 7.52

output@V[1,1]

## [1] 1

output@V[2,1]

## [1] -1.05189

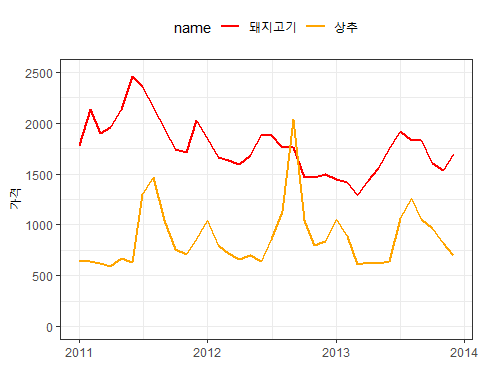
* r <= 1이라는 귀무가설 하에서 10% 유의수준으로 teststat(8.14)>cval(10pct)(7.52)이면 공적분관계가 있다고 본다.
* [1,1]의 값 1.00000과 [2,1]의 값 -1.05189의 곱이 음수이므로 양의 공적분 관계가 있다고 본다.

## 데이터 시각화

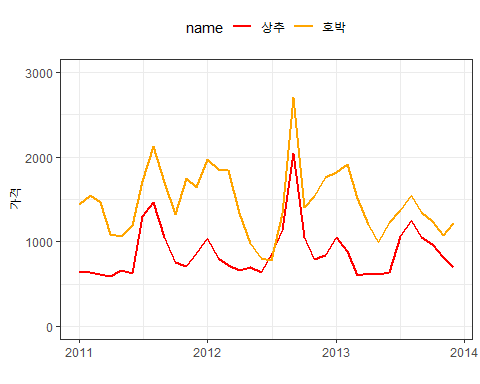
month.item.mean$month <- as.Date(as.yearmon(month.item.mean$month, "%Y-%m"))

p1 <- ggplot(month.item.mean[month.item.mean$name %in% c("돼지고기", "상추"),], aes(x=month, y=mean.price, colour=name, group=name)) +  
 geom\_line() + scale\_y\_continuous(name="가격", limits=c(0,2500)) +  
 theme\_bw() + xlab("")  
  
p2 <- ggplot(month.item.mean[month.item.mean$name %in% c("상추","호박"),], aes(x=month, y=mean.price, colour=name, group=name)) +  
 geom\_line() + scale\_y\_continuous(name="가격", limits=c(0,3000)) +  
 theme\_bw() + xlab("")  
  
month.item.mean[month.item.mean$name %in% c("상추"),]$mean.price <- month.item.mean[month.item.mean$name %in% c("상추"),]$mean.price+5000  
  
p4 <- ggplot(month.item.mean[month.item.mean$name %in% c("닭고기", "상추"),], aes(x=month, y=mean.price, colour=name, group=name)) +  
 geom\_line() + scale\_y\_continuous(name="가격", limits=c(5000,8000)) +  
 theme\_bw() + xlab("")  
  
month.item.mean[month.item.mean$name %in% c("호박"),]$mean.price <- month.item.mean[month.item.mean$name %in% c("호박"),]$mean.price+5000  
  
p5 <- ggplot(month.item.mean[month.item.mean$name %in% c("닭고기", "호박"),], aes(x=month, y=mean.price, colour=name, group=name)) +  
 geom\_line() + scale\_y\_continuous(name="가격", limits=c(5000,8000)) +  
 theme\_bw() + xlab("")

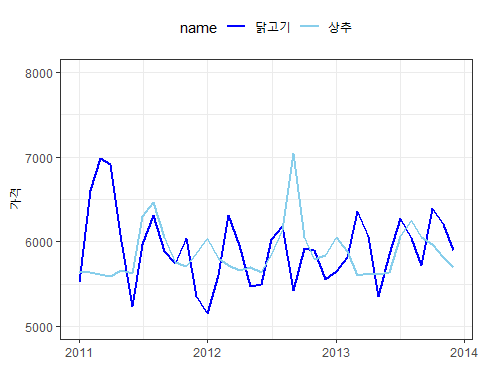
p1 + theme(legend.position = "top") + scale\_color\_manual(values = c("red", "orange")) +  
 geom\_line(size=1.0)



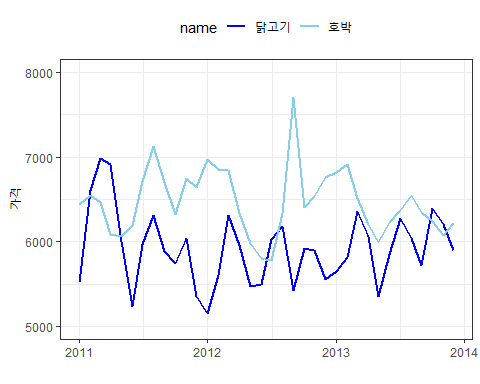
p2 + theme(legend.position = "top") + scale\_color\_manual(values = c("red", "orange")) + geom\_line(size=1.0)



p4 + theme(legend.position = "top") + scale\_colour\_manual(values = c("blue", "skyblue")) +  
 geom\_line(size=1.0)



p5 + theme(legend.position = "top") + scale\_color\_manual(values = c("blue", "skyblue")) +  
 geom\_line(size=1.0)



## 데이터 저장하기

write.csv(date.item.mean, "data.item.mean.csv", fileEncoding = "UTF-8")  
write.csv(month.item.mean, "month.item.mean.csv", fileEncoding = "UTF-8")

# 클러스터링 기법을 이용한 농축산물 데이터 분석

## 필요 패키지 및 데이터 불러오기

library3 <- c("plyr", "TSclust", "zoo", "ggplot2")  
unlist(lapply(library3, require, character.only=TRUE))

## Loading required package: TSclust

## Loading required package: wmtsa

## Loading required package: pdc

## Loading required package: cluster

## [1] TRUE TRUE TRUE TRUE

# install.packages("TSclust")  
  
pig.region <- read.csv("pig.region.csv", header = T, fileEncoding = "UTF-8")[,-1]  
head(pig.region, 10)

## 서울 부산 대구 인천 광주 대전 울산 수원 청주 전주 제주  
## 1 1738.333 1937.5 1627.5 1280 1610 1335 1980 2050 1890 1380 1980  
## 2 1738.333 1937.5 1627.5 1280 1610 1560 1980 2050 1890 1380 1980  
## 3 1738.333 1937.5 1627.5 1280 1610 1560 1980 2050 2010 1380 1980  
## 4 1538.333 1692.5 1442.5 1380 1390 1310 1000 2050 2010 1380 1980  
## 5 1538.333 1692.5 1492.5 1380 1390 1310 1000 2050 2010 1380 2140  
## 6 1543.333 1692.5 1492.5 1380 1390 1310 1000 2110 2340 1380 2140  
## 7 1543.333 1692.5 1492.5 1380 1390 1310 1000 1580 2340 1380 2140  
## 8 1543.333 1692.5 1492.5 1380 1390 1310 1000 1580 2340 1380 2140  
## 9 1676.667 2080.0 1760.0 1380 1635 1660 2380 1580 2340 1380 2140  
## 10 1676.667 2080.0 1860.0 1380 1735 1635 1380 1580 2340 1380 2320

pig.region.monthly.mean <- read.csv("pig.region.monthly.mean.csv", header = T, fileEncoding = "UTF-8")[,-1]  
head(pig.region.monthly.mean, 20)

## name region month mean.price  
## 1 광주 2401 2011-01-01 1782.619  
## 2 광주 2401 2011-02-01 2073.235  
## 3 광주 2401 2011-03-01 1771.364  
## 4 광주 2401 2011-04-01 1785.000  
## 5 광주 2401 2011-05-01 1956.750  
## 6 광주 2401 2011-06-01 2357.857  
## 7 광주 2401 2011-07-01 2452.976  
## 8 광주 2401 2011-08-01 2253.182  
## 9 광주 2401 2011-09-01 1997.375  
## 10 광주 2401 2011-10-01 1731.375  
## 11 광주 2401 2011-11-01 1676.136  
## 12 광주 2401 2011-12-01 2060.952  
## 13 광주 2401 2012-01-01 1810.250  
## 14 광주 2401 2012-02-01 1713.968  
## 15 광주 2401 2012-03-01 1697.143  
## 16 광주 2401 2012-04-01 1645.000  
## 17 광주 2401 2012-05-01 1658.730  
## 18 광주 2401 2012-06-01 1765.500  
## 19 광주 2401 2012-07-01 1815.152  
## 20 광주 2401 2012-08-01 1728.561

date.item.mean <- read.csv("data.item.mean.csv", header = T, fileEncoding = "UTF-8")[-1]  
head(date.item.mean, 10)

## item date mean.price code exp name  
## 1 111 2011-01-03 40953.33 1200 품목코드 쌀  
## 2 111 2011-01-04 40993.33 1200 품목코드 쌀  
## 3 111 2011-01-05 41153.33 1200 품목코드 쌀  
## 4 111 2011-01-06 41020.00 1200 품목코드 쌀  
## 5 111 2011-01-07 41013.33 1200 품목코드 쌀  
## 6 111 2011-01-10 41180.00 1200 품목코드 쌀  
## 7 111 2011-01-11 41080.00 1200 품목코드 쌀  
## 8 111 2011-01-12 41146.67 1200 품목코드 쌀  
## 9 111 2011-01-13 41146.67 1200 품목코드 쌀  
## 10 111 2011-01-14 41146.67 1200 품목코드 쌀

month.item.mean <- read.csv("month.item.mean.csv", header = T, fileEncoding = "UTF-8")[,-1]  
head(month.item.mean, 10)

## name item month mean.price  
## 1 닭고기 515 2011-01-01 5517.809  
## 2 닭고기 515 2011-02-01 6612.114  
## 3 닭고기 515 2011-03-01 6983.201  
## 4 닭고기 515 2011-04-01 6911.101  
## 5 닭고기 515 2011-05-01 6004.349  
## 6 닭고기 515 2011-06-01 5242.317  
## 7 닭고기 515 2011-07-01 5971.546  
## 8 닭고기 515 2011-08-01 6314.089  
## 9 닭고기 515 2011-09-01 5890.317  
## 10 닭고기 515 2011-10-01 5751.018

## 데이터 가공하기

temp <- dlply(date.item.mean, .(name), summarise, mean.price)  
farm.product <- data.frame("쌀"=unlist(temp$"쌀"), "배추"=unlist(temp$"배추"), "상추"=unlist(temp$"상추"), "호박"=unlist(temp$"호박"), "양파"=unlist(temp$"양파"), "파프리카"=unlist(temp$"파프리카"), "참깨"=unlist(temp$"참깨"), "사과"=unlist(temp$"사과"))  
head(farm.product)

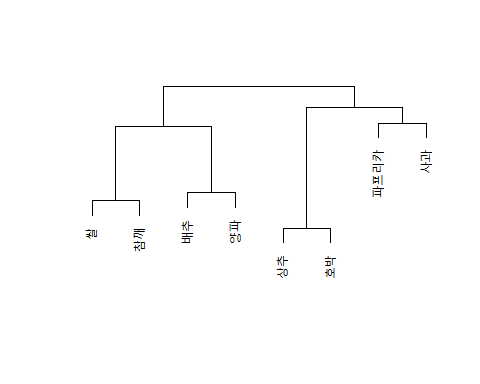
## 쌀 배추 상추 호박 양파 파프리카 참깨 사과  
## ..11 40953.33 4251.818 642.9024 1123.889 1903.333 1015.296 12133.33 24904.00  
## ..12 40993.33 4168.182 633.3095 1134.259 1877.667 1015.107 12133.33 24904.00  
## ..13 41153.33 4375.833 629.5227 1145.370 1864.333 1015.107 12133.33 25176.73  
## ..14 41020.00 4609.167 651.7826 1231.321 2070.000 1043.107 12133.33 25358.55  
## ..15 41013.33 4609.167 653.9565 1227.925 2071.667 1043.107 12133.33 25994.91  
## ..16 41180.00 4399.167 648.2391 1275.849 2075.000 1022.750 12133.33 26176.73

## 클러스터링 분석 및 데이터 시각화

### 농산물 자료

#### 군집분석

plot(hclust(diss(farm.product, "COR")), axes = F, ann = F)

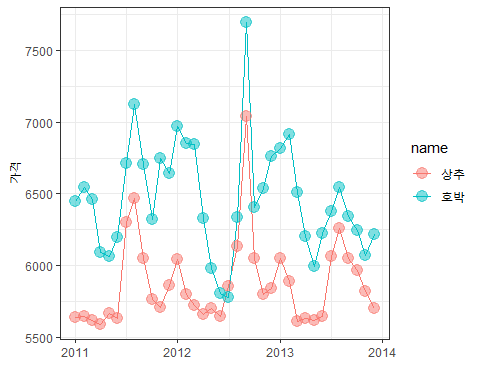


#### 시계열도 그리기 전 month 데이터 가공

month.item.mean$month <- as.Date(as.yearmon(month.item.mean$month, "%Y-%m"))

#### 가장 유의한 군집으로 형성된 상추와 호박에 대한 시계열 그림

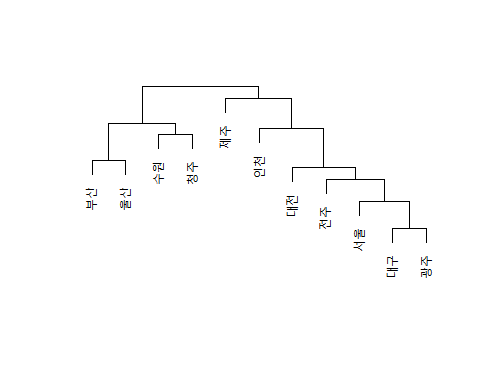
ggplot(month.item.mean[month.item.mean$name %in% c("상추", "호박"),], aes(x=month, y=mean.price, colour=name, group=name)) + geom\_line() + theme\_bw() + geom\_point(size=6, shape=20, alpha=0.5) + ylab("가격") + xlab("")



### 돼지고기 자료

#### 군집분석

plot(hclust(diss(pig.region, "COR")), axes = F, ann = F)

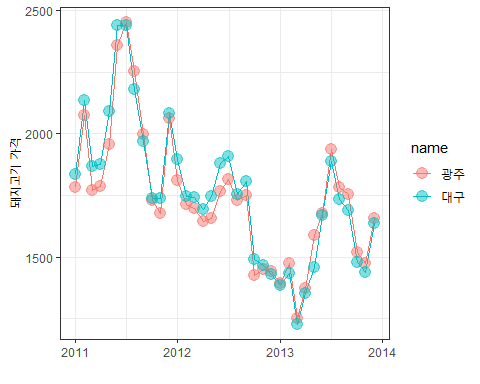


#### 월별 데이터 가공

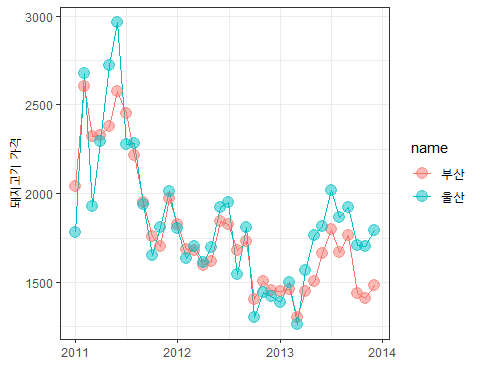
pig.region.monthly.mean$month <- as.Date(as.yearmon(pig.region.monthly.mean$month, "%Y-%m"))

#### 대구-광주, 부산-울산 시계열 그림

ggplot(pig.region.monthly.mean[pig.region.monthly.mean$region %in% c(2200, 2401),], aes(x=month, y=mean.price, colour=name, group=name)) +  
 geom\_line() + geom\_point(size=6, shape=20, alpha=0.5) + theme\_bw() +  
 ylab("돼지고기 가격") + xlab("")

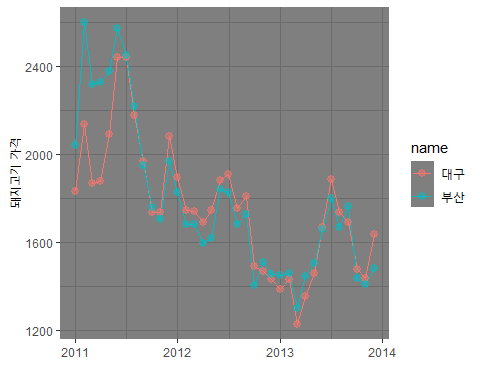


ggplot(pig.region.monthly.mean[pig.region.monthly.mean$region %in% c(2100, 2601),], aes(x=month, y=mean.price, colour=name, group=name)) +  
 geom\_line() +  
 geom\_point(size=6, shape=20, alpha=0.5) +  
 theme\_bw() +  
 xlab("") +  
 ylab("돼지고기 가격")



#### 다른 그룹인 대구-부산 시계열 그림

ggplot(pig.region.monthly.mean[pig.region.monthly.mean$name %in% c("대구", "부산"),], aes(x=month, y=mean.price, colour=name, group=name)) +  
 geom\_line() +  
 geom\_point(size=4, shape=20, alpha=0.5) +  
 theme\_dark() +  
 ylab("돼지고기 가격") +  
 xlab("")



구제역 발생 후 대구는 큰 영향을 받았으나 부산은 그렇지 않았기 때문으로 추정됨.

# 날씨 자료와 농산물 자료의 인과관계 분석

## 필요 패키지 및 데이터 불러오기

library4 <- c("plyr", "stringr", "dygraphs", "zoo", "xts")  
# install.packages("dygraphs")  
# install.packages("xts")  
unlist(lapply(library4, require, character.only=TRUE))

## Loading required package: dygraphs

## Loading required package: xts

## [1] TRUE TRUE TRUE TRUE TRUE

product <- read.csv("product.csv", header = T, fileEncoding = "UTF-8")  
weather <- read.csv("weather.csv", header = T, fileEncoding = "EUC-KR")  
code <- read.csv("code.csv", header = T, fileEncoding = "EUC-KR")

## 데이터 가공하기

### 농축산물 데이터 가공

#### 지역코드 추출

subset(code, code$"구분코드설명"=="지역코드")

## 구분코드 구분코드설명 분류코드 분류코드설명  
## 16 1300 지역코드 1101 서울   
## 17 1300 지역코드 2100 부산  
## 18 1300 지역코드 2200 대구  
## 19 1300 지역코드 2300 인천  
## 20 1300 지역코드 2401 광주  
## 21 1300 지역코드 2501 대전  
## 22 1300 지역코드 2601 울산  
## 23 1300 지역코드 3111 수원  
## 24 1300 지역코드 3113 의정부  
## 25 1300 지역코드 3145 용인  
## 26 1300 지역코드 3211 춘천  
## 27 1300 지역코드 3311 청주  
## 28 1300 지역코드 3511 전주  
## 29 1300 지역코드 3613 순천  
## 30 1300 지역코드 3711 포항  
## 31 1300 지역코드 3714 안동  
## 32 1300 지역코드 3814 창원  
## 33 1300 지역코드 3911 제주

#### 품목별 코드번호: category 오브젝트 생성

category <- subset(code, code$"구분코드설명" == "품목코드")  
category

## 구분코드 구분코드설명 분류코드 분류코드설명  
## 6 1200 품목코드 111 쌀  
## 7 1200 품목코드 211 배추  
## 8 1200 품목코드 214 상추  
## 9 1200 품목코드 224 호박  
## 10 1200 품목코드 245 양파  
## 11 1200 품목코드 256 파프리카  
## 12 1200 품목코드 312 참깨   
## 13 1200 품목코드 411 사과  
## 14 1200 품목코드 514 돼지고기  
## 15 1200 품목코드 515 닭고기

#### 변수명 한글 -> 영어

colnames(product) <- c('date', 'category', 'item', 'region', 'mart', 'price')  
  
colnames(category) <- c('code', 'exp', 'item', 'name')  
  
head(product)

## date category item region mart price  
## 1 2011-01-03 100 111 3511 350401 39600  
## 2 2011-01-03 100 111 3311 330401 40000  
## 3 2011-01-03 100 111 1101 110251 37000  
## 4 2011-01-03 100 111 1101 110401 43900  
## 5 2011-01-03 100 111 1101 110402 39800  
## 6 2011-01-03 100 111 1101 110403 39600

head(category)

## code exp item name  
## 6 1200 품목코드 111 쌀  
## 7 1200 품목코드 211 배추  
## 8 1200 품목코드 214 상추  
## 9 1200 품목코드 224 호박  
## 10 1200 품목코드 245 양파  
## 11 1200 품목코드 256 파프리카

#### 서울지역의 가격만 추출하여 품몸(item), 일자(date)별로 평균 가격을 구하여 품목에 대한 데이터 category데이터와 merge하여 seoul.item 데이터 생성

mean.price.new <- ddply(product[which(product$region==1101),],.(item, date), summarise, mean.price=mean(price))  
  
seoul.item <- merge(mean.price.new, category, by="item", all = T)  
  
head(seoul.item, 10)

## item date mean.price code exp name  
## 1 111 2011-01-03 41016.67 1200 품목코드 쌀  
## 2 111 2011-01-04 41016.67 1200 품목코드 쌀  
## 3 111 2011-01-05 41350.00 1200 품목코드 쌀  
## 4 111 2011-01-06 41350.00 1200 품목코드 쌀  
## 5 111 2011-01-07 41316.67 1200 품목코드 쌀  
## 6 111 2011-01-10 41316.67 1200 품목코드 쌀  
## 7 111 2011-01-11 41316.67 1200 품목코드 쌀  
## 8 111 2011-01-12 41316.67 1200 품목코드 쌀  
## 9 111 2011-01-13 41316.67 1200 품목코드 쌀  
## 10 111 2011-01-14 41316.67 1200 품목코드 쌀

#### 일별 평균가격을 생성하여 품목별로 정렬한 seoul.item.mean 데이터 생성

seoul.item.mean <- ddply(seoul.item, .(item, date), summarise, item.name=name, mean.price=mean.price)  
head(seoul.item.mean, 10)

## item date item.name mean.price  
## 1 111 2011-01-03 쌀 41016.67  
## 2 111 2011-01-04 쌀 41016.67  
## 3 111 2011-01-05 쌀 41350.00  
## 4 111 2011-01-06 쌀 41350.00  
## 5 111 2011-01-07 쌀 41316.67  
## 6 111 2011-01-10 쌀 41316.67  
## 7 111 2011-01-11 쌀 41316.67  
## 8 111 2011-01-12 쌀 41316.67  
## 9 111 2011-01-13 쌀 41316.67  
## 10 111 2011-01-14 쌀 41316.67

### 기상데이터 가공

str(weather)

## 'data.frame': 691920 obs. of 4 variables:  
## $ 지역 : Factor w/ 93 levels "강릉(청)","강진군(공)",..: 27 27 27 27 27 27 27 27 27 27 ...  
## $ 기상구분: Factor w/ 2 levels "강수량","평균기온": 1 1 1 1 1 1 1 1 1 1 ...  
## $ 측정값 : num NA 3 NA 8.5 37.5 NA NA 3 0.4 NA ...  
## $ 일자 : Factor w/ 1860 levels "2010-01-01","2010-01-02",..: 157 188 219 250 281 312 343 3 34 65 ...

head(weather)

## 지역 기상구분 측정값 일자  
## 1 백령도(기) 강수량 NA 2010-06-02  
## 2 백령도(기) 강수량 3.0 2010-07-02  
## 3 백령도(기) 강수량 NA 2010-08-02  
## 4 백령도(기) 강수량 8.5 2010-09-02  
## 5 백령도(기) 강수량 37.5 2010-10-02  
## 6 백령도(기) 강수량 NA 2010-11-02

#### 변수명 한글 -> 영어

colnames(weather) <- c('region', 'category', 'value', 'date')

#### weather 데이터를 지역별 이름순으로 나누어 region.weather 리스트 형식의 데이터 생성

region.weather <- dlply(weather, .(region))  
names(region.weather)

## [1] "강릉(청)" "강진군(공)" "강화(관)" "거제(관)" "거창(기)"   
## [6] "경주(공)" "고산(기)" "고창(구)" "고흥(관)" "광양(공)"   
## [11] "광주(청)" "구미(기)" "군산(기)" "금산(관)" "김해시(공)"  
## [16] "남원(기)" "남해(관)" "대관령(기)" "대구(구)" "대구(기)"   
## [21] "대전(청)" "동두천(기)" "동해(기)" "목포(기)" "문경(관)"   
## [26] "밀양(관)" "백령도(기)" "보령(기)" "보성군(공)" "보은(관)"   
## [31] "봉화(관)" "부산(청)" "부안(관)" "부여(관)" "북강릉(청)"  
## [36] "북창원(공)" "산청(관)" "상주(기)" "서귀포(기)" "서산(기)"   
## [41] "서울(청)" "성산(기)" "속초(기)" "수원(기)" "순창(공)"   
## [46] "순천(구)" "순천(기)" "안동(기)" "양산(공)" "양평(관)"   
## [51] "여수(기)" "영광(공)" "영덕(관)" "영월(기)" "영주(관)"   
## [56] "영천(관)" "완도(기)" "울릉도(기)" "울산(기)" "울진(기)"   
## [61] "원주(기)" "의령군(공)" "의성(관)" "이천(기)" "인제(관)"   
## [66] "인천(기)" "임실(관)" "장수(관)" "장흥(관)" "전주(기)"   
## [71] "정선군(공)" "정읍(기)" "제주(청)" "제천(관)" "진도군(공)"  
## [76] "진주(기)" "창원(기)" "천안(기)" "철원(기)" "청송군(공)"  
## [81] "청주(기)" "추풍령(기)" "춘천(기)" "충주(기)" "태백(관)"   
## [86] "통영(기)" "파주(기)" "포항(기)" "함양군(공)" "합천(관)"   
## [91] "해남(관)" "홍천(관)" "흑산도(기)"

#### 서울에 대한 기상데이터 확인

head(region.weather[[41]], 10)

## region category value date  
## 1 서울(청) 평균기온 -6.8 2011-01-01  
## 2 서울(청) 평균기온 -0.2 2011-02-01  
## 3 서울(청) 평균기온 0.5 2011-03-01  
## 4 서울(청) 평균기온 9.1 2011-04-01  
## 5 서울(청) 평균기온 12.5 2011-05-01  
## 6 서울(청) 평균기온 18.0 2011-06-01  
## 7 서울(청) 평균기온 25.1 2011-07-01  
## 8 서울(청) 평균기온 25.6 2011-08-01  
## 9 서울(청) 평균기온 27.0 2011-09-01  
## 10 서울(청) 평균기온 12.7 2011-10-01

tail(region.weather[[41]], 10)

## region category value date  
## 7431 서울(청) 강수량 9.5 2010-03-31  
## 7432 서울(청) 강수량 NA 2010-04-31  
## 7433 서울(청) 강수량 1.5 2010-05-31  
## 7434 서울(청) 강수량 NA 2010-06-31  
## 7435 서울(청) 강수량 NA 2010-07-31  
## 7436 서울(청) 강수량 2.0 2010-08-31  
## 7437 서울(청) 강수량 NA 2010-09-31  
## 7438 서울(청) 강수량 NA 2010-10-31  
## 7439 서울(청) 강수량 NA 2010-11-31  
## 7440 서울(청) 강수량 NA 2010-12-31

#### 서울 강수량 데이터

init.seoul.rain <- region.weather[[41]][which(region.weather[[41]][,2]=="강수량"),]  
  
head(init.seoul.rain, 10)

## region category value date  
## 373 서울(청) 강수량 NA 2011-01-01  
## 374 서울(청) 강수량 NA 2011-02-01  
## 375 서울(청) 강수량 2.3 2011-03-01  
## 376 서울(청) 강수량 NA 2011-04-01  
## 377 서울(청) 강수량 0.0 2011-05-01  
## 378 서울(청) 강수량 21.5 2011-06-01  
## 379 서울(청) 강수량 NA 2011-07-01  
## 380 서울(청) 강수량 0.5 2011-08-01  
## 381 서울(청) 강수량 NA 2011-09-01  
## 382 서울(청) 강수량 NA 2011-10-01

tail(init.seoul.rain, 10)

## region category value date  
## 7431 서울(청) 강수량 9.5 2010-03-31  
## 7432 서울(청) 강수량 NA 2010-04-31  
## 7433 서울(청) 강수량 1.5 2010-05-31  
## 7434 서울(청) 강수량 NA 2010-06-31  
## 7435 서울(청) 강수량 NA 2010-07-31  
## 7436 서울(청) 강수량 2.0 2010-08-31  
## 7437 서울(청) 강수량 NA 2010-09-31  
## 7438 서울(청) 강수량 NA 2010-10-31  
## 7439 서울(청) 강수량 NA 2010-11-31  
## 7440 서울(청) 강수량 NA 2010-12-31

init.seoul.rain[which(init.seoul.rain$date=='2010-01-02'),]

## region category value date  
## 5593 서울(청) 강수량 1.4 2010-01-02  
## 7081 서울(청) 강수량 1.4 2010-01-02

#### 날짜순 배열

sort.seoul.rain <- dlply(init.seoul.rain, .(date))  
head(sort.seoul.rain)

## $`2010-01-01`  
## region category value date  
## 1 서울(청) 강수량 NA 2010-01-01  
## 2 서울(청) 강수량 NA 2010-01-01  
##   
## $`2010-01-02`  
## region category value date  
## 1 서울(청) 강수량 1.4 2010-01-02  
## 2 서울(청) 강수량 1.4 2010-01-02  
##   
## $`2010-01-03`  
## region category value date  
## 1 서울(청) 강수량 NA 2010-01-03  
## 2 서울(청) 강수량 NA 2010-01-03  
##   
## $`2010-01-04`  
## region category value date  
## 1 서울(청) 강수량 14.2 2010-01-04  
## 2 서울(청) 강수량 14.2 2010-01-04  
##   
## $`2010-01-05`  
## region category value date  
## 1 서울(청) 강수량 0 2010-01-05  
## 2 서울(청) 강수량 0 2010-01-05  
##   
## $`2010-01-06`  
## region category value date  
## 1 서울(청) 강수량 NA 2010-01-06  
## 2 서울(청) 강수량 NA 2010-01-06

#### 일별로 데이터값들이 2개씩 중복되어 있으므로 하나의 값만 추출하여 resort.seoul.rain데이터 생성

resort.seoul.rain <- lapply(1:length(sort.seoul.rain), function(x) sort.seoul.rain[[x]][1,])  
  
head(resort.seoul.rain, 10)

## [[1]]  
## region category value date  
## 1 서울(청) 강수량 NA 2010-01-01  
##   
## [[2]]  
## region category value date  
## 1 서울(청) 강수량 1.4 2010-01-02  
##   
## [[3]]  
## region category value date  
## 1 서울(청) 강수량 NA 2010-01-03  
##   
## [[4]]  
## region category value date  
## 1 서울(청) 강수량 14.2 2010-01-04  
##   
## [[5]]  
## region category value date  
## 1 서울(청) 강수량 0 2010-01-05  
##   
## [[6]]  
## region category value date  
## 1 서울(청) 강수량 NA 2010-01-06  
##   
## [[7]]  
## region category value date  
## 1 서울(청) 강수량 NA 2010-01-07  
##   
## [[8]]  
## region category value date  
## 1 서울(청) 강수량 NA 2010-01-08  
##   
## [[9]]  
## region category value date  
## 1 서울(청) 강수량 0.4 2010-01-09  
##   
## [[10]]  
## region category value date  
## 1 서울(청) 강수량 NA 2010-01-10

#### resort.seoul.rain 데이터에서 date와 value 값만 추출하여 seoul.rain데이터 생성

seoul.rain <- data.frame(date=unlist(lapply(1:length(resort.seoul.rain), function(x) resort.seoul.rain[[x]][,4])),rain=unlist(lapply(1:length(resort.seoul.rain),function(x) resort.seoul.rain[[x]][,3])))  
  
head(seoul.rain, 10)

## date rain  
## 1 2010-01-01 NA  
## 2 2010-01-02 1.4  
## 3 2010-01-03 NA  
## 4 2010-01-04 14.2  
## 5 2010-01-05 0.0  
## 6 2010-01-06 NA  
## 7 2010-01-07 NA  
## 8 2010-01-08 NA  
## 9 2010-01-09 0.4  
## 10 2010-01-10 NA

## NA -> 0

seoul.rain[,2][is.na(seoul.rain[,2])] <- 0  
head(seoul.rain, 10)

## date rain  
## 1 2010-01-01 0.0  
## 2 2010-01-02 1.4  
## 3 2010-01-03 0.0  
## 4 2010-01-04 14.2  
## 5 2010-01-05 0.0  
## 6 2010-01-06 0.0  
## 7 2010-01-07 0.0  
## 8 2010-01-08 0.0  
## 9 2010-01-09 0.4  
## 10 2010-01-10 0.0

### 농산물 데이터와 기상 데이터의 가공

str(seoul.item.mean)

## 'data.frame': 7449 obs. of 4 variables:  
## $ item : int 111 111 111 111 111 111 111 111 111 111 ...  
## $ date : Factor w/ 745 levels "2011-01-03","2011-01-04",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ item.name : Factor w/ 81 levels "AA유통","AB유통",..: 63 63 63 63 63 63 63 63 63 63 ...  
## $ mean.price: num 41017 41017 41350 41350 41317 ...

str(seoul.rain)

## 'data.frame': 1860 obs. of 2 variables:  
## $ date: Factor w/ 1860 levels "2010-01-01","2010-01-02",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ rain: num 0 1.4 0 14.2 0 0 0 0 0.4 0 ...

head(seoul.item.mean)

## item date item.name mean.price  
## 1 111 2011-01-03 쌀 41016.67  
## 2 111 2011-01-04 쌀 41016.67  
## 3 111 2011-01-05 쌀 41350.00  
## 4 111 2011-01-06 쌀 41350.00  
## 5 111 2011-01-07 쌀 41316.67  
## 6 111 2011-01-10 쌀 41316.67

tail(seoul.item.mean)

## item date item.name mean.price  
## 7444 515 2013-12-23 닭고기 5417.143  
## 7445 515 2013-12-24 닭고기 5417.143  
## 7446 515 2013-12-26 닭고기 5241.667  
## 7447 515 2013-12-27 닭고기 5241.667  
## 7448 515 2013-12-30 닭고기 5473.333  
## 7449 515 2013-12-31 닭고기 5473.333

head(seoul.rain)

## date rain  
## 1 2010-01-01 0.0  
## 2 2010-01-02 1.4  
## 3 2010-01-03 0.0  
## 4 2010-01-04 14.2  
## 5 2010-01-05 0.0  
## 6 2010-01-06 0.0

tail(seoul.rain)

## date rain  
## 1855 2014-12-26 0  
## 1856 2014-12-27 0  
## 1857 2014-12-28 0  
## 1858 2014-12-29 0  
## 1859 2014-12-30 0  
## 1860 2014-12-31 0

#### 농산물(2011~2013), 기상(2010~2014) 이므로 date에 따라 두 데이터를 합쳐서 seoul.item.rain생성

seoul.item.rain <- merge(seoul.rain, seoul.item.mean, by="date", all.y = T)  
  
head(seoul.item.rain, 20)

## date rain item item.name mean.price  
## 1 2011-01-03 0 111 쌀 41016.667  
## 2 2011-01-03 0 256 파프리카 990.000  
## 3 2011-01-03 0 224 호박 1150.000  
## 4 2011-01-03 0 214 상추 625.000  
## 5 2011-01-03 0 211 배추 4940.000  
## 6 2011-01-03 0 312 참깨 10700.000  
## 7 2011-01-03 0 411 사과 20000.000  
## 8 2011-01-03 0 514 돼지고기 1738.333  
## 9 2011-01-03 0 515 닭고기 4980.000  
## 10 2011-01-03 0 245 양파 1851.667  
## 11 2011-01-04 0 256 파프리카 990.000  
## 12 2011-01-04 0 224 호박 1136.667  
## 13 2011-01-04 0 312 참깨 10700.000  
## 14 2011-01-04 0 111 쌀 41016.667  
## 15 2011-01-04 0 245 양파 1856.667  
## 16 2011-01-04 0 514 돼지고기 1738.333  
## 17 2011-01-04 0 211 배추 4230.000  
## 18 2011-01-04 0 515 닭고기 5686.667  
## 19 2011-01-04 0 411 사과 20000.000  
## 20 2011-01-04 0 214 상추 625.000

str(seoul.item.rain)

## 'data.frame': 7449 obs. of 5 variables:  
## $ date : Factor w/ 1860 levels "2010-01-01","2010-01-02",..: 375 375 375 375 375 375 375 375 375 375 ...  
## $ rain : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ item : int 111 256 224 214 211 312 411 514 515 245 ...  
## $ item.name : Factor w/ 81 levels "AA유통","AB유통",..: 63 79 81 58 55 72 57 54 51 65 ...  
## $ mean.price: num 41017 990 1150 625 4940 ...

seoul.item.rain <- merge(seoul.rain, seoul.item.mean, by="date", all = T)  
  
head(seoul.item.rain)

## date rain item item.name mean.price  
## 1 2010-01-01 0.0 NA <NA> NA  
## 2 2010-01-02 1.4 NA <NA> NA  
## 3 2010-01-03 0.0 NA <NA> NA  
## 4 2010-01-04 14.2 NA <NA> NA  
## 5 2010-01-05 0.0 NA <NA> NA  
## 6 2010-01-06 0.0 NA <NA> NA

seoul.item.rain <- ddply(seoul.item.rain[!seoul.item.rain$mean.price %in% NA,], .(item.name))  
  
head(seoul.item.rain)

## date rain item item.name mean.price  
## 1 2011-01-03 0 515 닭고기 4980.000  
## 2 2011-01-04 0 515 닭고기 5686.667  
## 3 2011-01-05 0 515 닭고기 5686.667  
## 4 2011-01-06 0 515 닭고기 6113.333  
## 5 2011-01-07 0 515 닭고기 5620.000  
## 6 2011-01-10 0 515 닭고기 5620.000

str(seoul.item.rain)

## 'data.frame': 7449 obs. of 5 variables:  
## $ date : Factor w/ 1860 levels "2010-01-01","2010-01-02",..: 375 376 377 378 379 382 383 384 385 386 ...  
## $ rain : num 0 0 0 0 0 0 1.9 0 0 1.1 ...  
## $ item : int 515 515 515 515 515 515 515 515 515 515 ...  
## $ item.name : Factor w/ 81 levels "AA유통","AB유통",..: 51 51 51 51 51 51 51 51 51 51 ...  
## $ mean.price: num 4980 5687 5687 6113 5620 ...

## 데이터 시각화

### plot()함수를 이용한 Graph Overlap

par(mar=c(3,5,3,5))  
plot(as.Date(seoul.item.rain[seoul.item.rain$item.name %in% c("호박"),]$date), seoul.item.rain[seoul.item.rain$item.name %in% c("호박"),]$mean.price, type="l", col="blue", xlab="", ylab="", ylim=c(0,4000))  
mtext("가격", side=2, line=3)  
  
par(new=T)  
plot(as.Date(seoul.item.rain[seoul.item.rain$item.name %in% c("상추"),]$date), seoul.item.rain[seoul.item.rain$item.name %in% c("상추"),]$mean.price, type="l", col="green", xlab="", ylab="", ylim=c(0,4000), axes=F)  
  
par(new=T)  
plot(as.Date(seoul.item.rain$date), seoul.item.rain$rain, type="l", col="red", xlab="", ylab="", ylim=c(0,400), axes=F)  
  
axis(4, ylim=c(0,400), col.axis="red", las=3)  
mtext("강수량", side=4, line=3)

