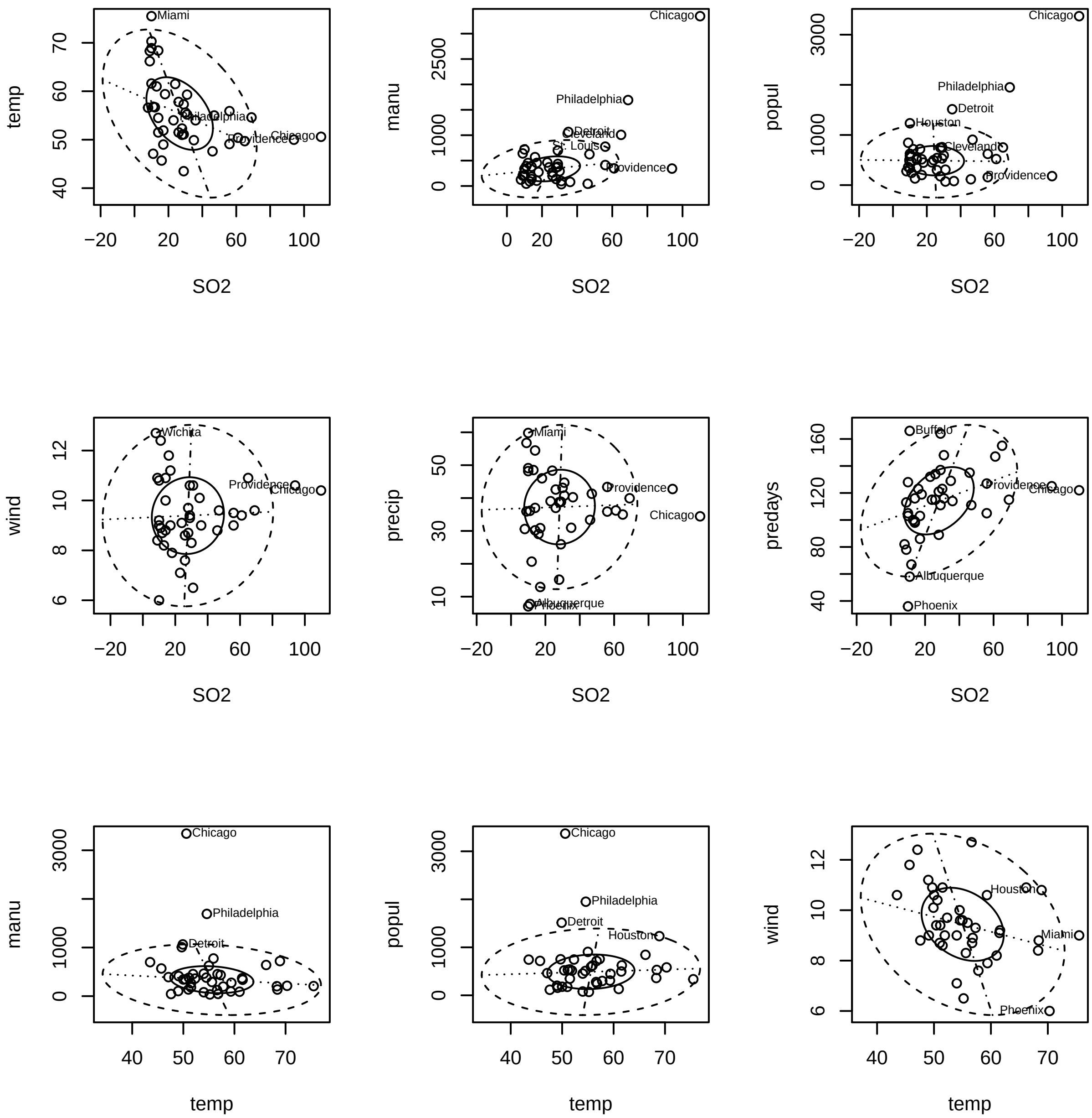
Homework 2

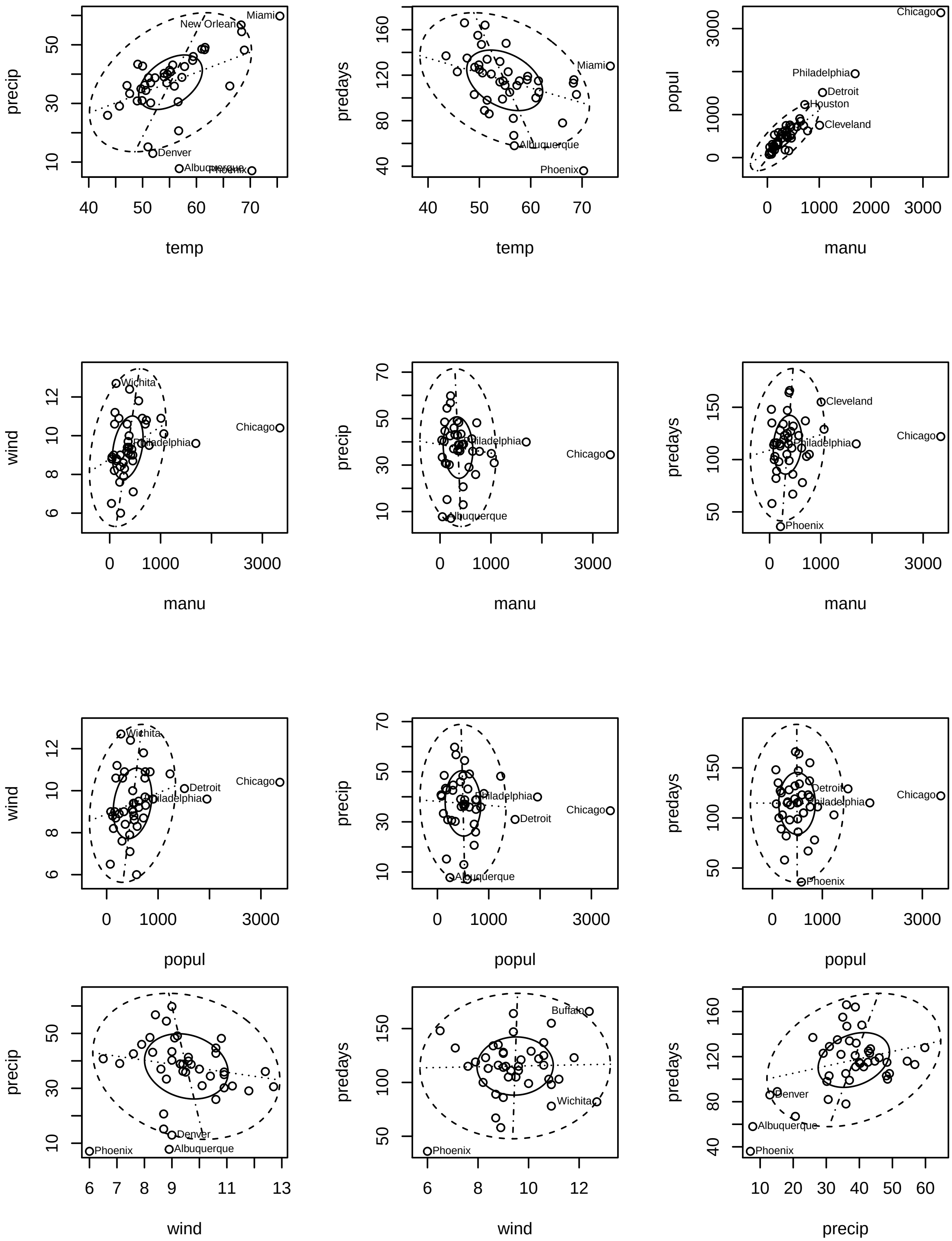
Cody Frisby

1/30/2017

2.1

Here I display all 21 of the bi-variate boxplots using the `bvplot` function from the `MVA` package. I also identify all points that are either on the outer ellipse or outside it.





Counting each time a city appears as an "outlier" on each of the plots we have the following results.

City	frequency
Chicago	15

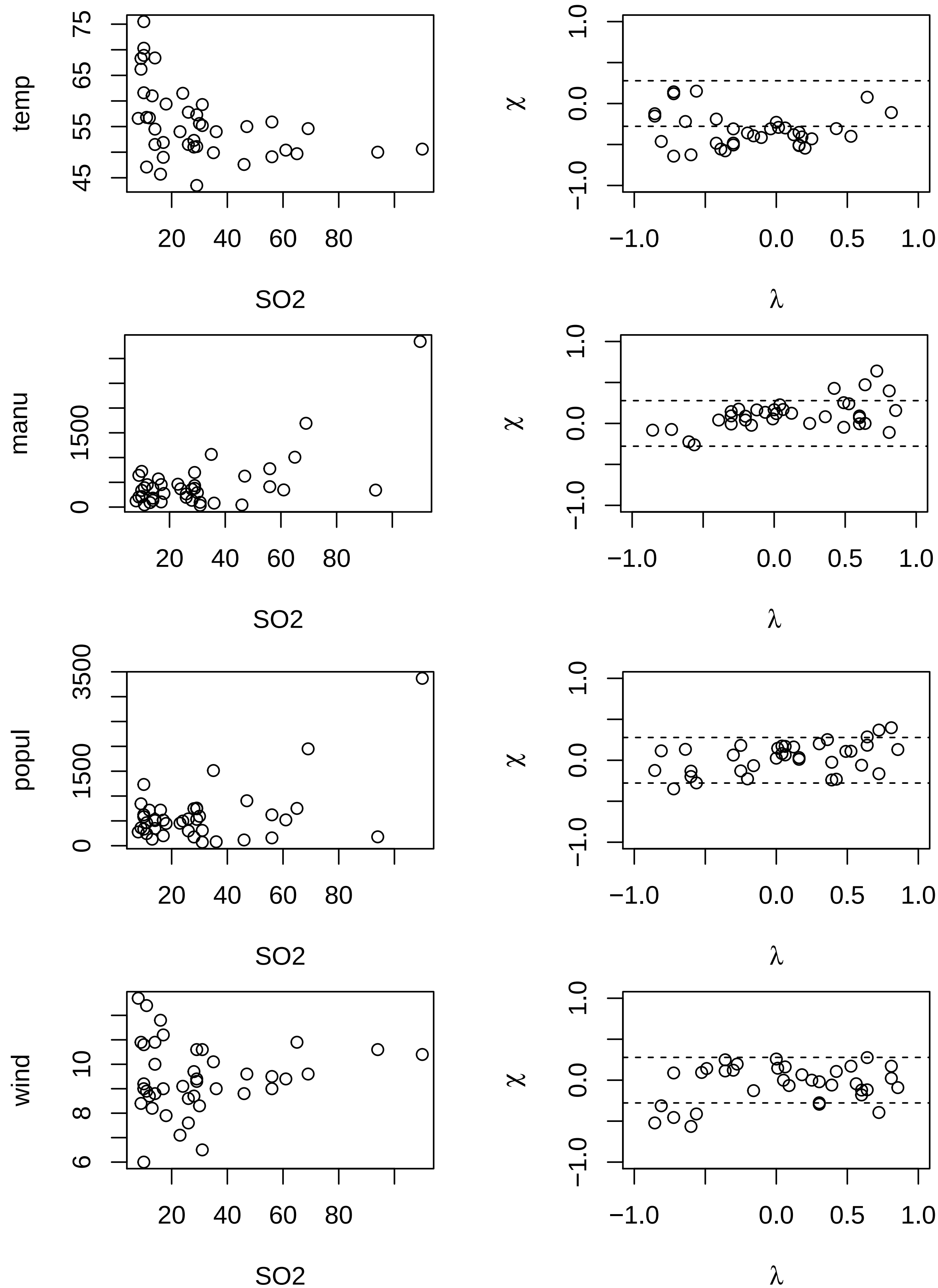
City	frequency
Philadelphia	12
Phoenix	10
Albuquerque	8
Detroit	8
Providence	6
Miami	5
Cleveland	4
Houston	4
Wichita	4
Denver	3
Buffalo	2
New Orlean	1
St. Louis	1

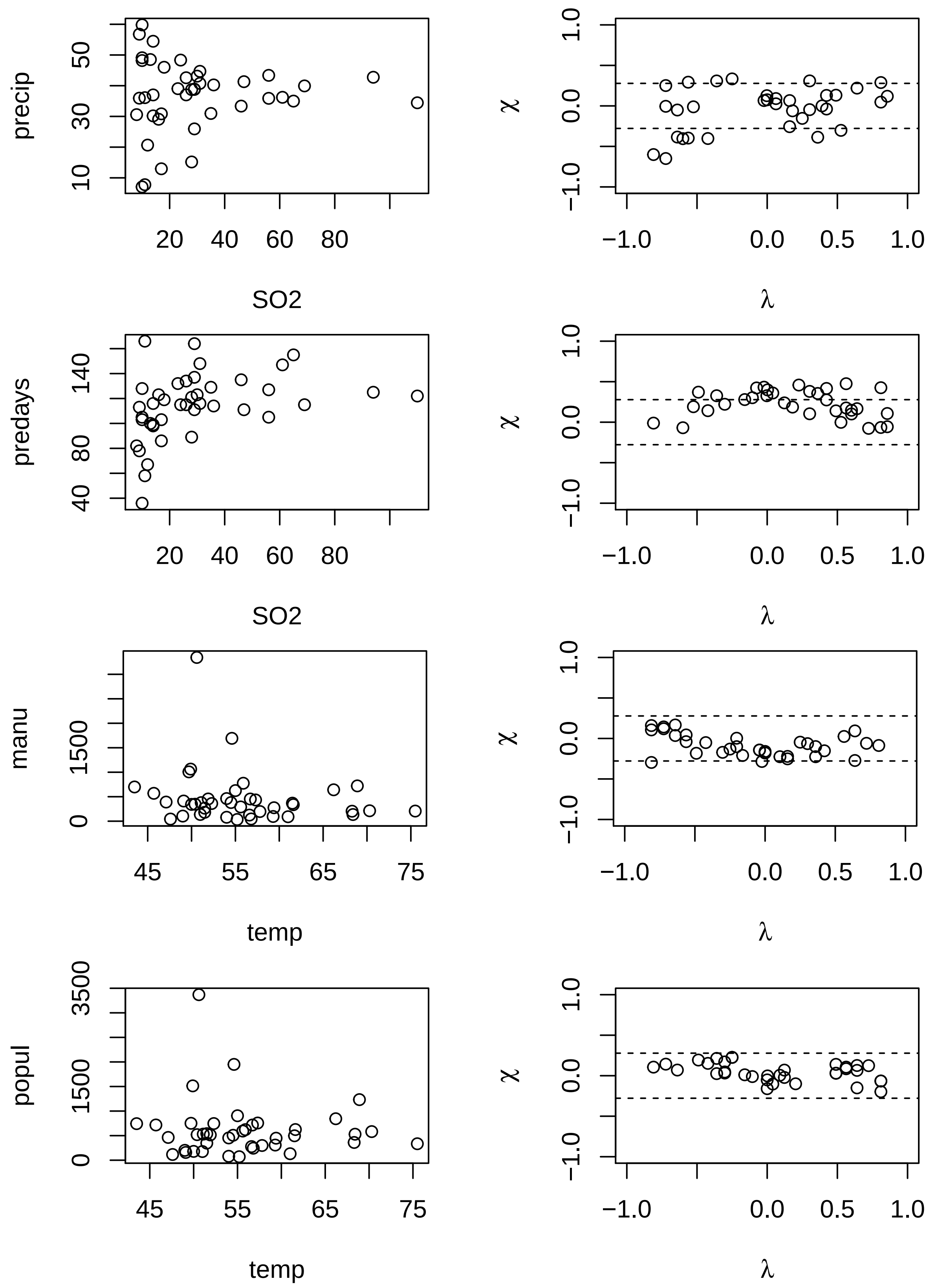
And below is displayed all the $2x2$ correlations with the observations that were on or outside the outer ellipse.

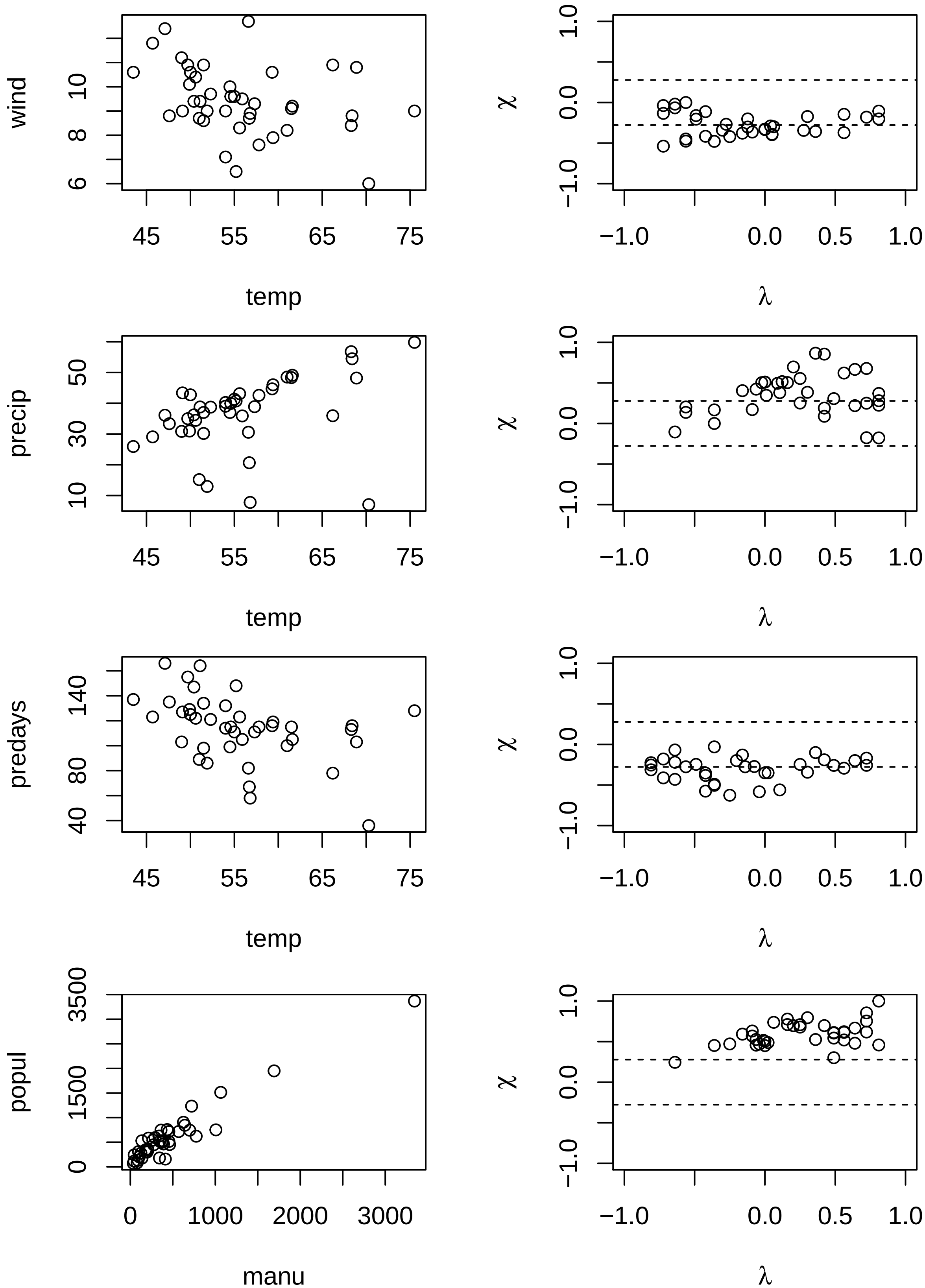
x	y	rho
SO2	temp	-0.4588760
SO2	manu	0.0201079
SO2	popul	-0.0431609
SO2	wind	0.0366234
SO2	precip	-0.0361940
SO2	predays	0.4984332
temp	manu	-0.1533296
temp	popul	0.0380325
temp	wind	-0.3440909
temp	precip	0.6061363
temp	predays	-0.4402388
manu	popul	0.7700381
manu	wind	0.4163601
manu	precip	-0.1443024
manu	predays	0.0770077
popul	wind	0.3090494
popul	precip	-0.0561544
popul	predays	-0.0278808
wind	precip	-0.3199555
wind	predays	-0.0686141
precip	predays	0.1693367

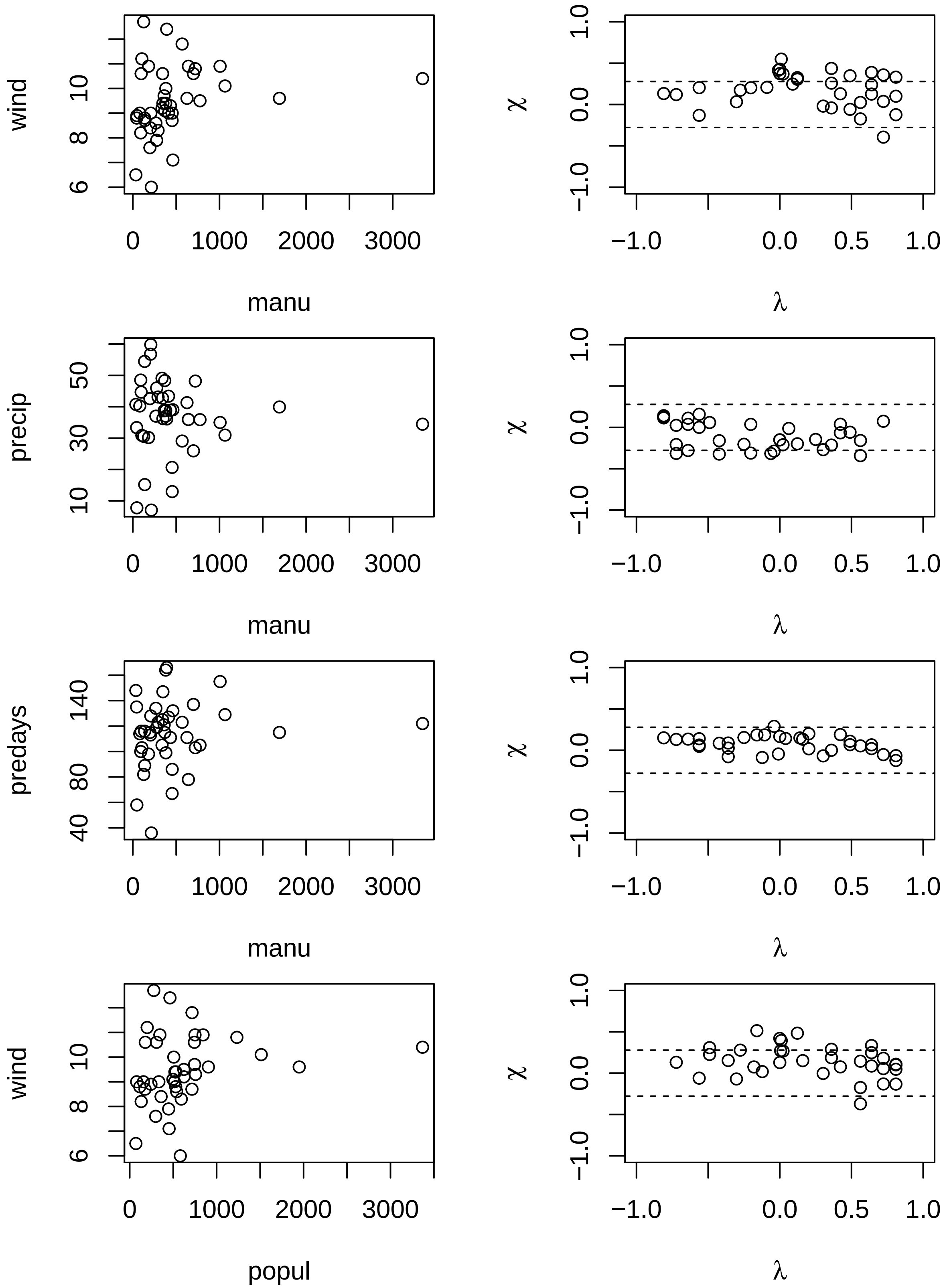
2.2

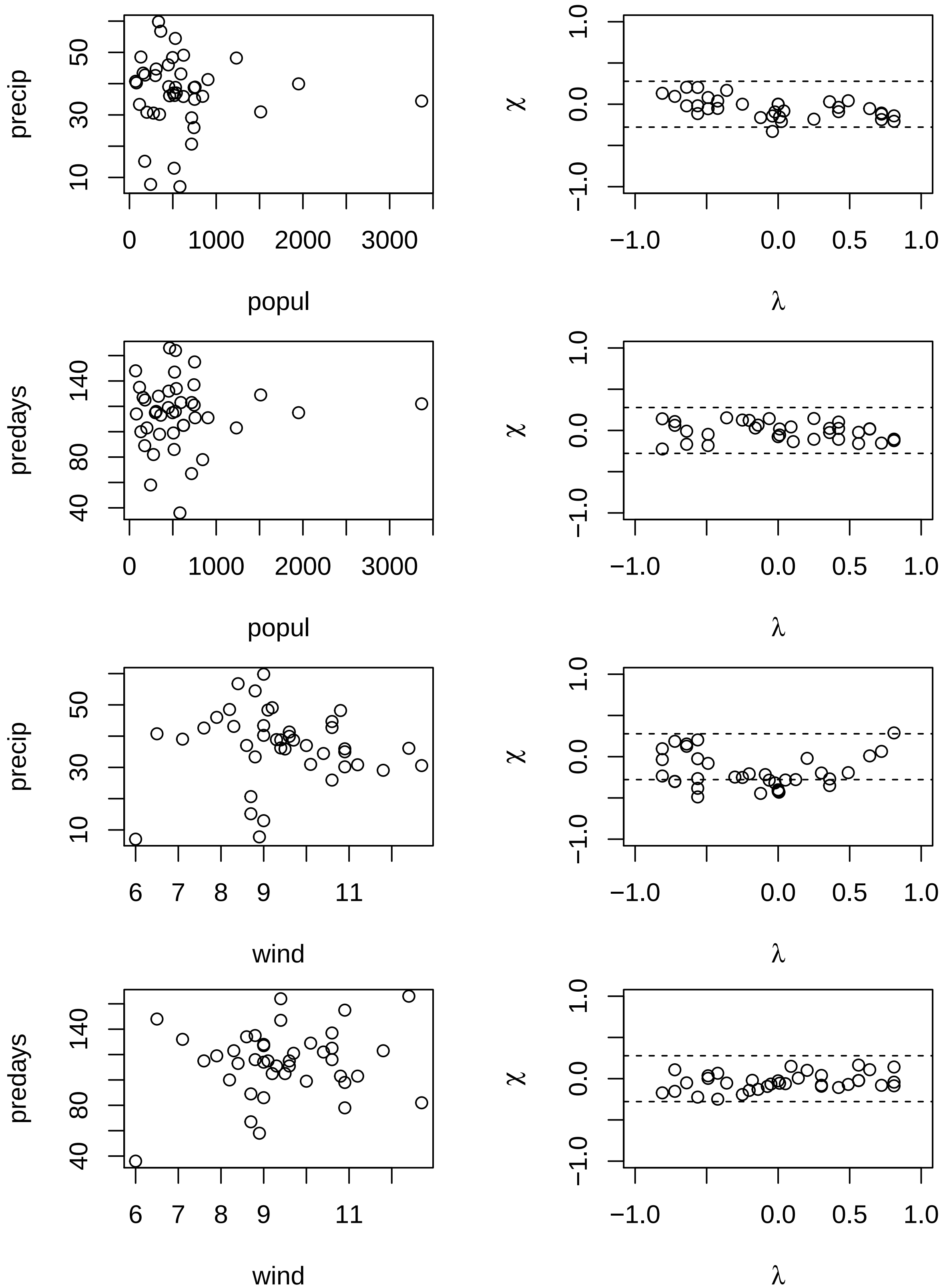
Using the method from the book, I display all bivariate chi-plots.

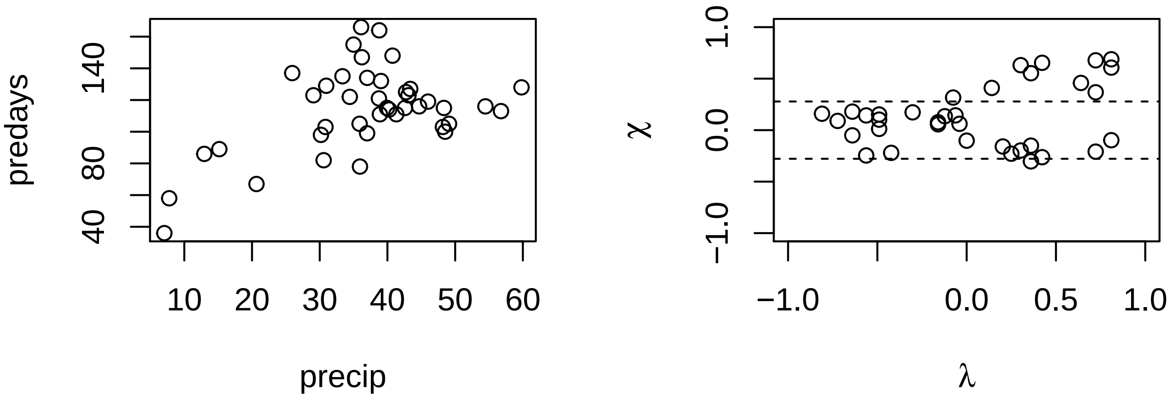

temp
manu
popul
wind
SO2
χ
λ


precip
10
30
50
20
40
60
80
SO2
1.0
0.0
−1.0
χ
−1.0
0.0
0.5
1.0
λ
predays
40
80
140
20
40
60
80
SO2
1.0
0.0
−1.0
χ
−1.0
0.0
0.5
1.0
λ
manu
0
1500
45
55
65
75
temp
1.0
0.0
−1.0
χ
−1.0
0.0
0.5
1.0
λ
popul
0
1500
3500
45
55
65
75
temp
1.0
0.0
−1.0
χ
−1.0
0.0
0.5
1.0
λ


wind
temp
precip
temp
predays
temp
popul
manu
χ
λ

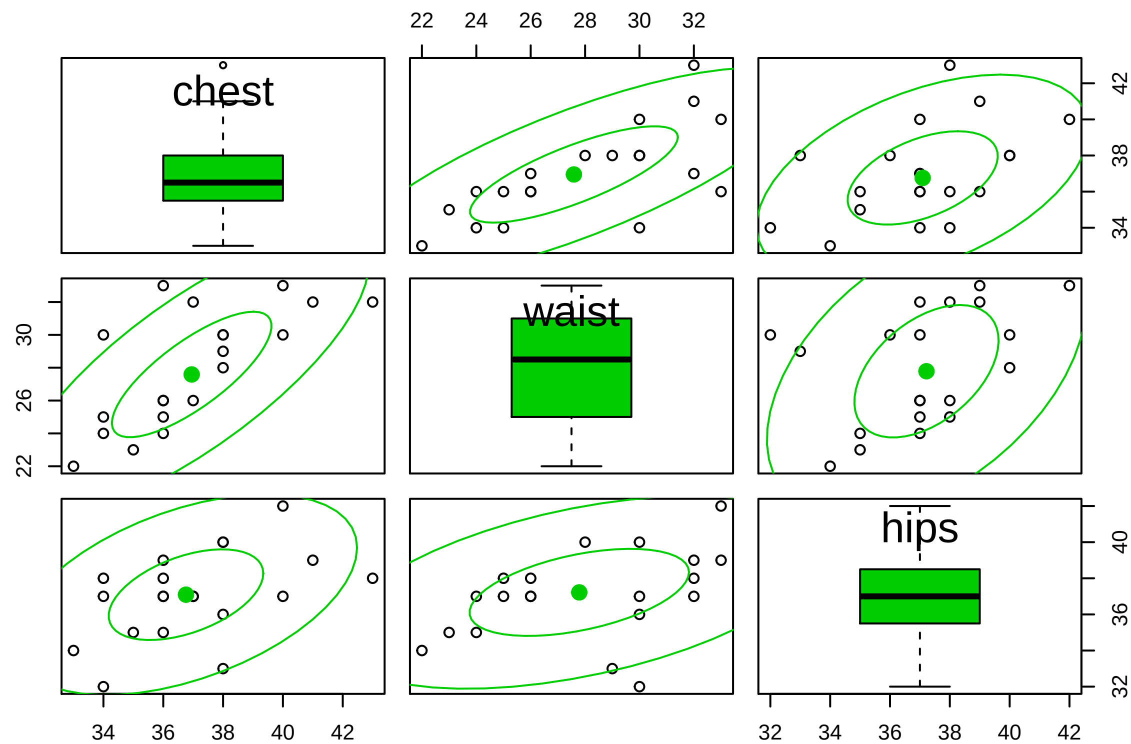

wind
manu
χ
λ
precip
manu
χ
λ
predays
manu
χ
λ
wind
popul
χ
λ


precip
popul
predays
wind
χ
λ



I think this method is very useful for quickly identifying covariates that appear to be independent of one another and also others that are highly dependent on one another. I like it.

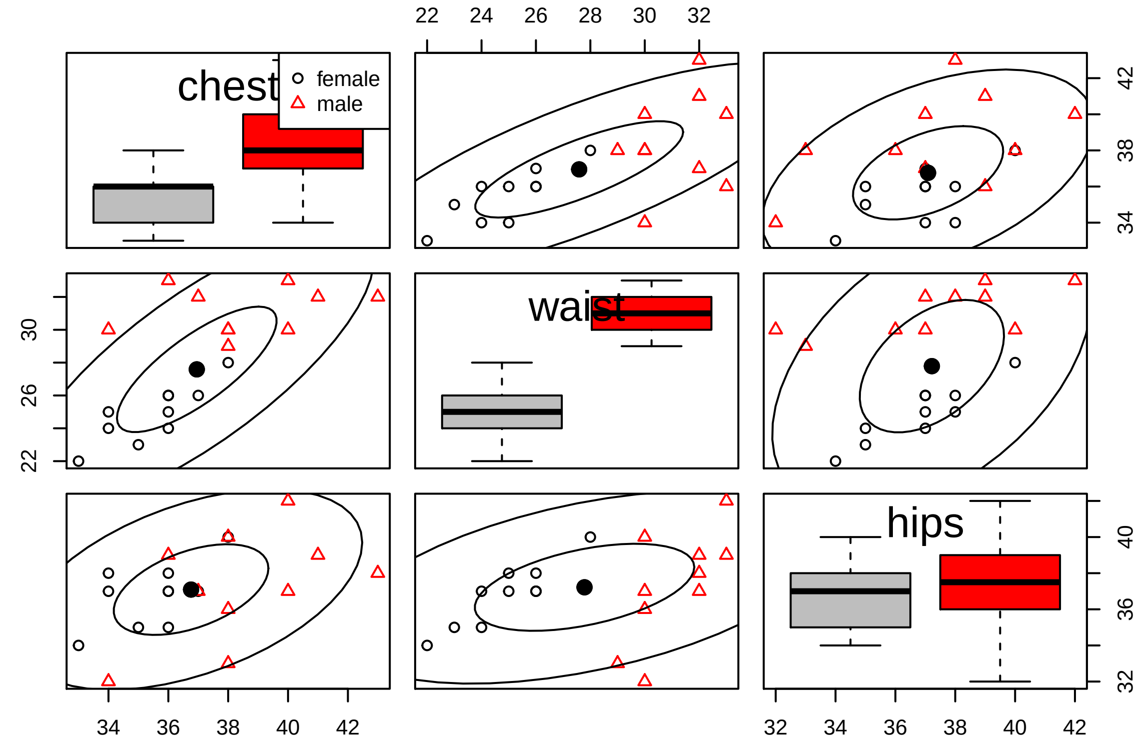
2.3



For bivariate continuous data, I find the simple scatterplot easier to look at and quickly interpret than the plot from **Fig. 2.17**. The contour plots, for me, take a little longer to look at.

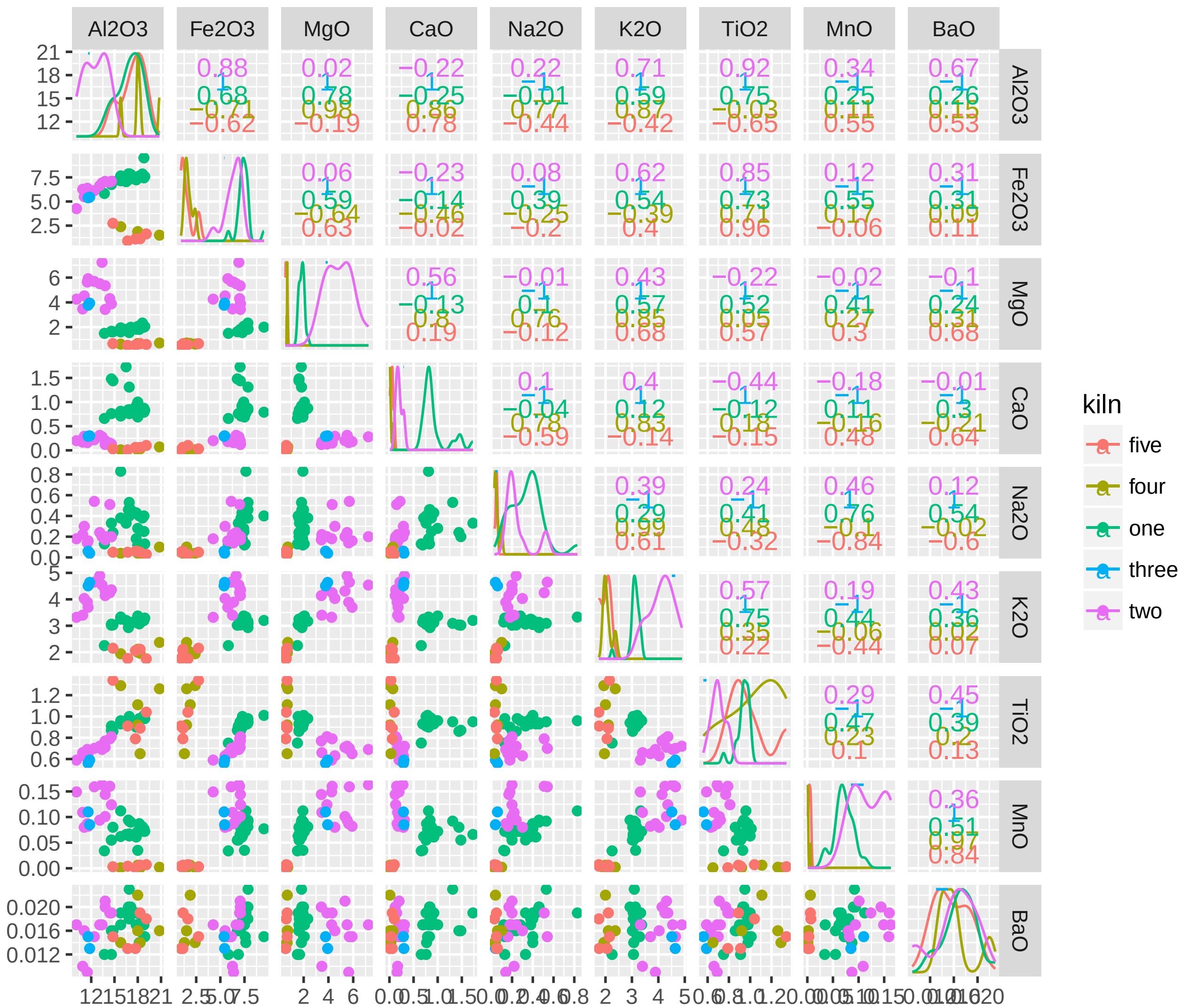
2.4

Plotting the scatterplot matrix with gender as a color overlay:

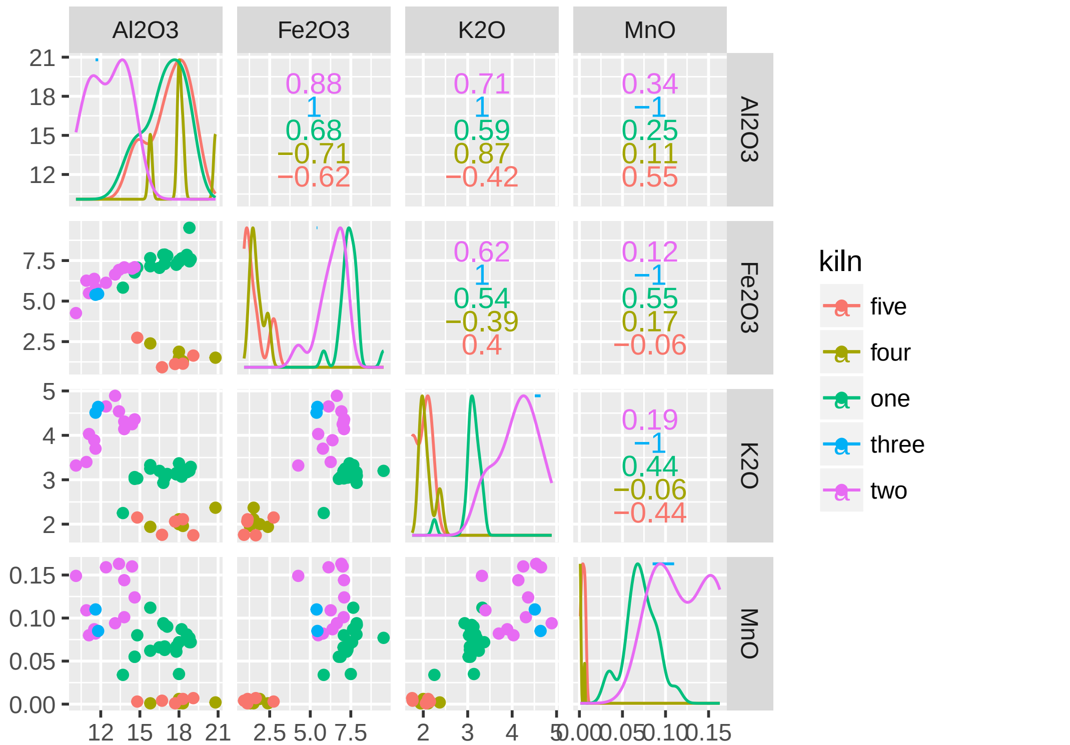


2.5

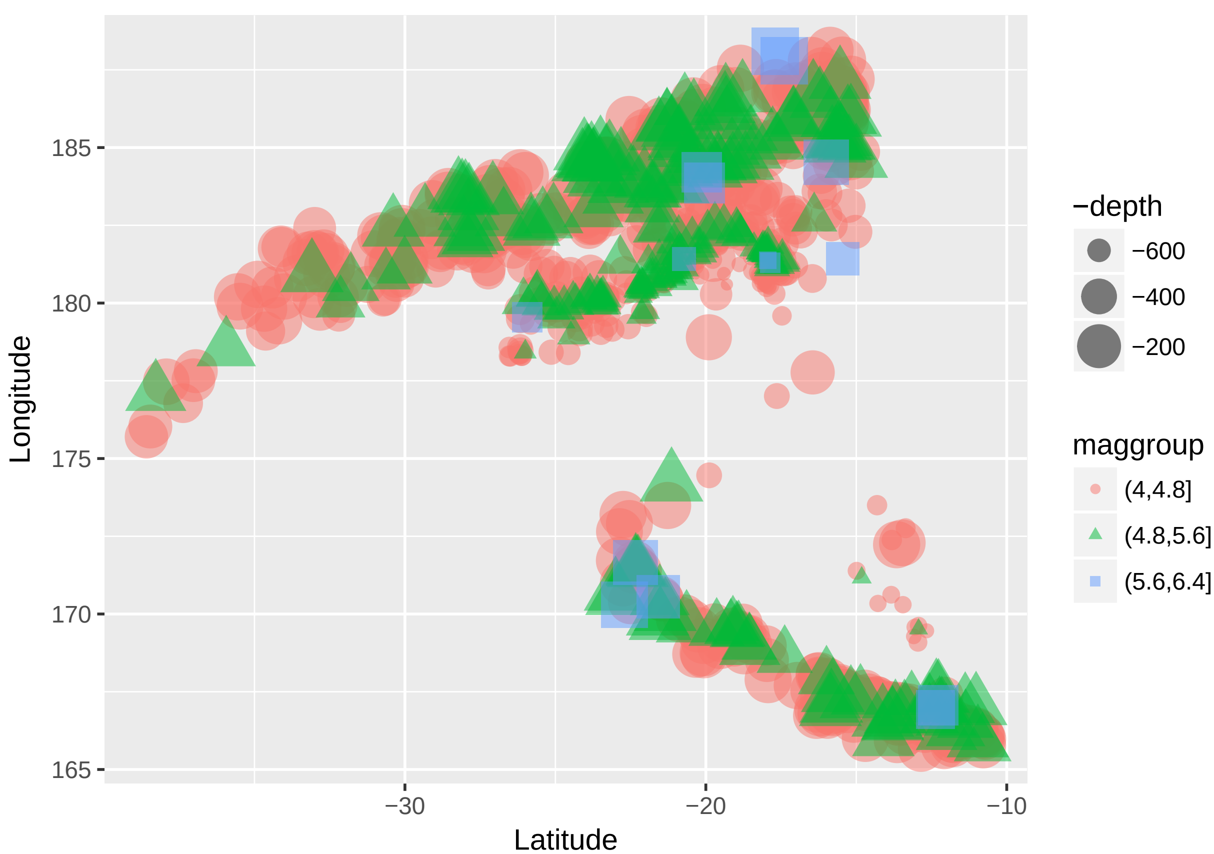
From the pottery data set, here are all the $2x2$ pairs with the kiln overlayed as the color.



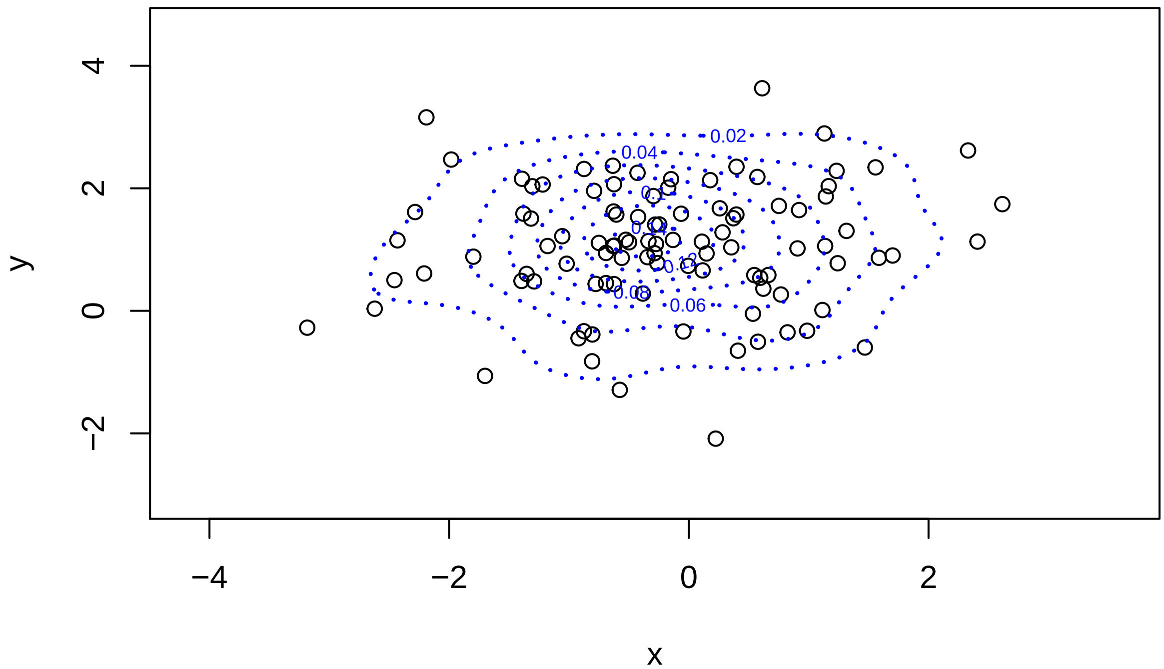
There appears to be some clustering of the data based on the kiln on some of the bivariate plots. By reducing the number of variables that are plotted we can see this a little more clearly.

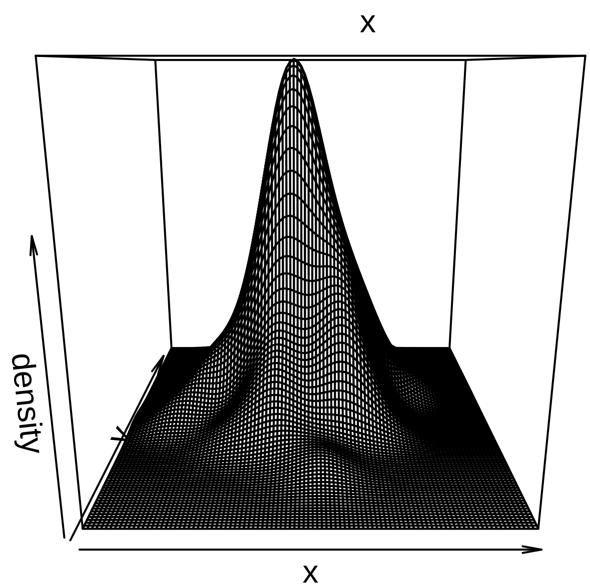

Al2O3
Fe2O3
K2O
MnO
0.88
1
0.68
−0.71
−0.62
0.71
1
0.59
0.87
−0.42
0.34
−1
0.25
0.11
0.55
0.62
1
0.54
−0.39
0.4
0.12
−1
0.55
0.17
−0.06
0.19
−1
0.44
−0.06
−0.44
kiln
five
four
one
three
two

2.6

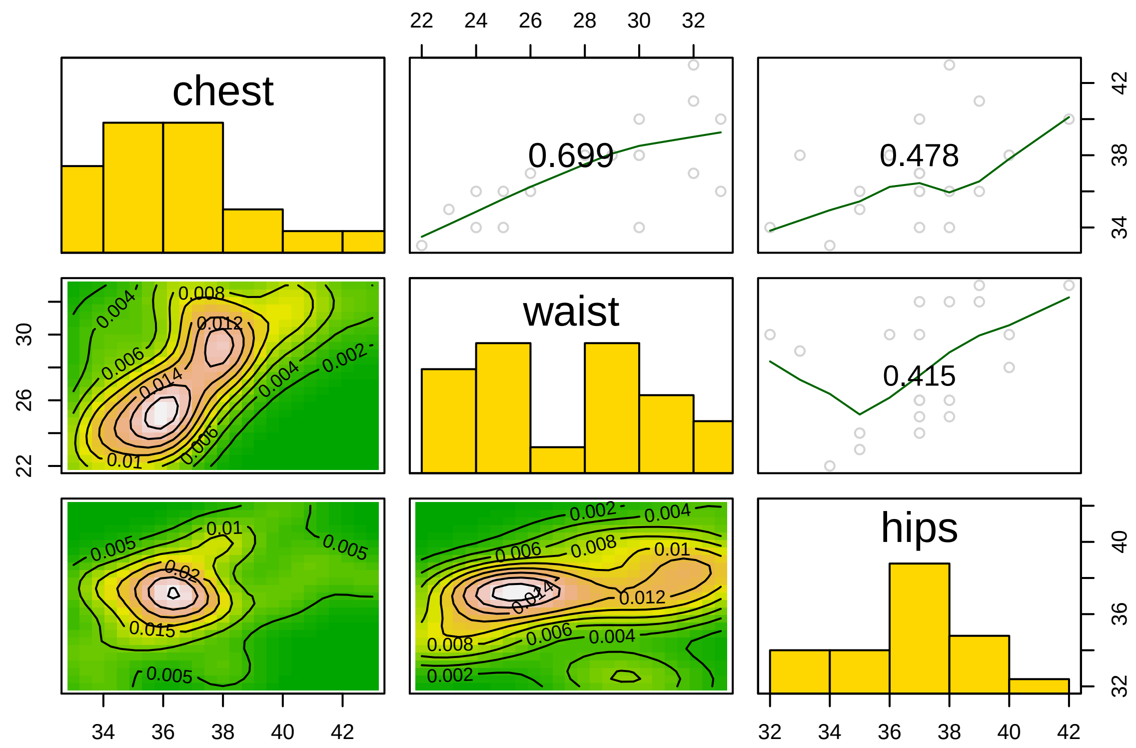

Longitude
Latitude
−depth
−600
−400
−200
maggroup
(4,4.8]
(4.8,5.6]
(5.6,6.4]

2.7





2.8



R Code:

```
### 2.1
# bivariate boxplot.
library(MVA)
df <- read.csv("~/Documents/STAT4400/data/USairpollution.csv")
id <- df$X
df <- df[-1]
row.names(df) <- id
combos <- t(combn(1:length(names(df)), 2))
par(mfrow = c(3,3))
outliers <- list()
# plot all 21 plots and store the outliers in a list
par(mfrow = c(3,3))
for (i in 1:21) { outliers[[i]] <- bvbox(df[, combos[i,]],
    xlab = names(df)[combos[i, ]][1],
    ylab = names(df)[combos[i, ]][2], labels = id)
}
## correlation with outliers removed.
corr <- list()
for (i in 1:21){
  corr[[i]] <- cor(df[!row.names(df) %in% names(outliers[[i]]),
                      combos[i, ]])
}
rho <- vector()
```

```
temp <- data.frame(x = vector(length = 21),
                   y = vector(length = 21))
for (i in 1:21){
  rho[i] <- corr[[i]][1,2]
  temp[i, ] <- dimnames(corr[[i]])[[1]]
}
df.cor <- data.frame(temp, rho)
knitr::kable(df.cor)
### 2.2
par(mfrow = c(1,2))
for (i in 1:21) {
  labs <- names(df)[combos[i, ]]
  plot(x = df[, combos[i, ][1]],
    y = df[, combos[i, ][2]], xlab = labs[1], ylab = labs[2])
  chiplot(x = df[, combos[i, ][1]],
    y = df[, combos[i, ][2]])
}
### 2.3
measure <- read.csv("~/Documents/STAT4400/data/measure.csv")
library(GGally)
ggpairs(measure)
## 2.4
ggpairs(measure, columns = 1:3, ggplot2::aes(colour = gender))
## 2.5
pot <- read.csv("~/Documents/STAT4400/data/pottery.csv")
pot <- pot[-1]
pot$kiln <- c(rep("one", 21), rep("two", 12), rep("three", 2),
                     rep("four", 5), rep("five", 5))

library(GGally)
pm <- ggscatmat(data = pot, columns = 1:9, color = "kiln")
pm # a somewhat slow plot.
pm <- ggscatmat(data = pot, columns = c(1,2,6,8),
                color = "kiln")
pm # this one is a lot quicker.
### 2.6
quakes <- read.csv("~/Documents/STAT4400/data/quakes.csv")
# divide the variable mag into three equal groups:
quakes$maggroup <- cut(sort(quakes$mag), breaks = 3)
# I like this better than the books example.
library(ggplot2)
gg <- ggplot(data = quakes, aes(lat, long, colour = maggroup,
                        size = -depth, shape = maggroup))
gg <- gg + geom_point(alpha = 1/2)
# change the size of the scale
gg <- gg + scale_size_continuous(range = c(1, 8))
gg <- gg + xlab("Latitude") + ylab("Longitude")
gg
##2.7
x <- rnorm(n = 100,mean = 0,sd = 1)
y <- rnorm(n = 100,mean = 1,sd = 1)
rand.data <- cbind(x, y)
rand.2d <- KernSmooth::bkde2D(x = rand.data,
```

```
                  bandwidth = c(.5,.5),
                  gridsize = c(100L,100L),
                  range.x = list(c(-4,4),c(-4,4)),truncate = T)
# plot the points
plot(rand.data, xlab = "x", ylab = "y",
     xlim = c(min(x) - 1, max(x) + 1),
     ylim = c(min(y) - 1, max(y) + 1))
# plot the contour lines
contour(x = rand.2d$x1, y = rand.2d$x2, z = rand.2d$fhat,add = T,
        lty = 3, col = "blue", lwd = 2)
# cool 3D plot
persp(x = rand.2d$x1, y = rand.2d$x2, z = rand.2d$fhat,
      xlab = "x", ylab = "y", zlab = "density")
## 2.8
# using the Resource Selection package.
ResourceSelection::kdepairs(measure[,-4])
```