HW4_Christianson_Ryan

Ryan Christianson

September 18, 2018

```r
# global settings
knitr::opts_chunk$set(warning = FALSE, wessage = FALSE, comment = NA)

# libraries
library("readxl")
library("ggplot2")
library("grid")
library("ggmosaic")
library("gridExtra")
library("GGally")
```

```r
# multiplot function from
# http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/

# Multiple plot function
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots / cols)),
                    ncol = cols, nrow = ceiling(numPlots / cols))
  }

 if (numPlots == 1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))
    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

Problem 1

Done.

Problem 2

Done.

Problem 3

He belives that EDA allows us to see what geneal treands are important in the data and which are not worth thinking any more about. It allows us to skip through some analyses that would not be successful.

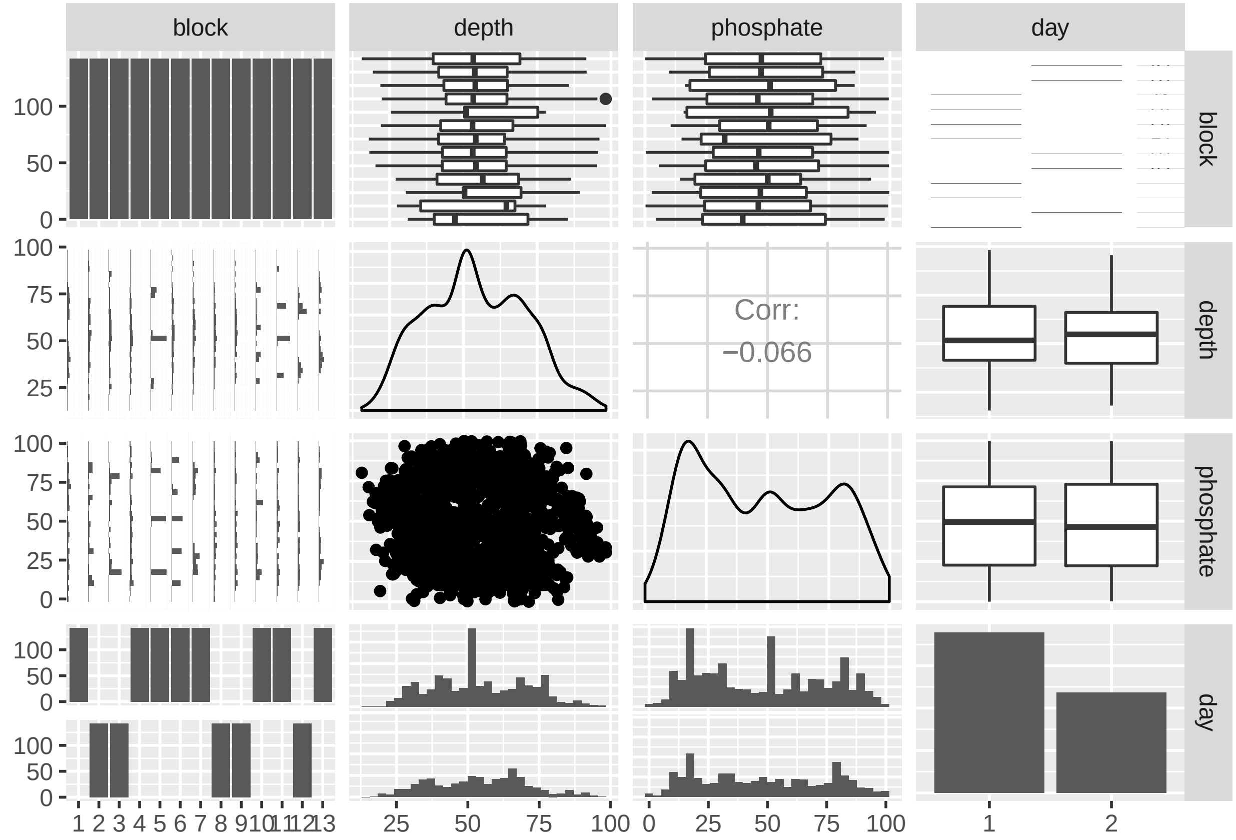
Problem 4

```
data.1 <- read_xlsx("HW4_data.xlsx", sheet = 1)
data.2 <- read_xlsx("HW4_data.xlsx", sheet = 2)
data.1$day <- 1
data.2$day <- 2
data <- rbind(data.1, data.2)

# block and day are factors
data$block <- as.factor(data$block)
data$day <- as.factor(data$day)

# summary statistics
summary(data)
```

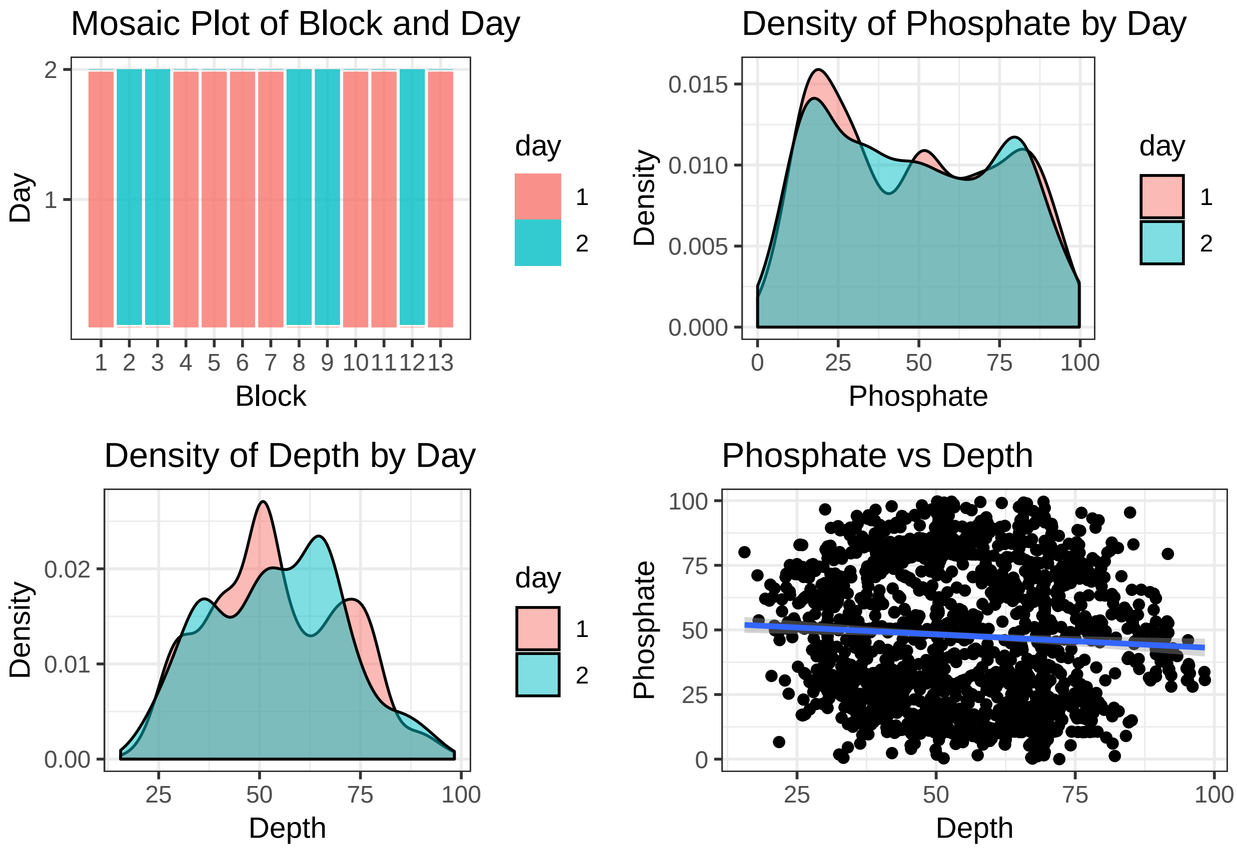
```
     block          depth          phosphate          day
 1      :142   Min.   :15.56   Min.   : 0.01512   1:1136
 2      :142   1st Qu.:41.07   1st Qu.:22.56107   2: 710
 3      :142   Median :52.59   Median :47.59445
 4      :142   Mean   :54.27   Mean   :47.83510
 5      :142   3rd Qu.:67.28   3rd Qu.:71.81078
 6      :142   Max.   :98.29   Max.   :99.69468
 (Other):994
```

```
# pairs plot of data
# ggpairs may fail if ggally is out of date due to updates to ggplot2
ggpairs(data)
```

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



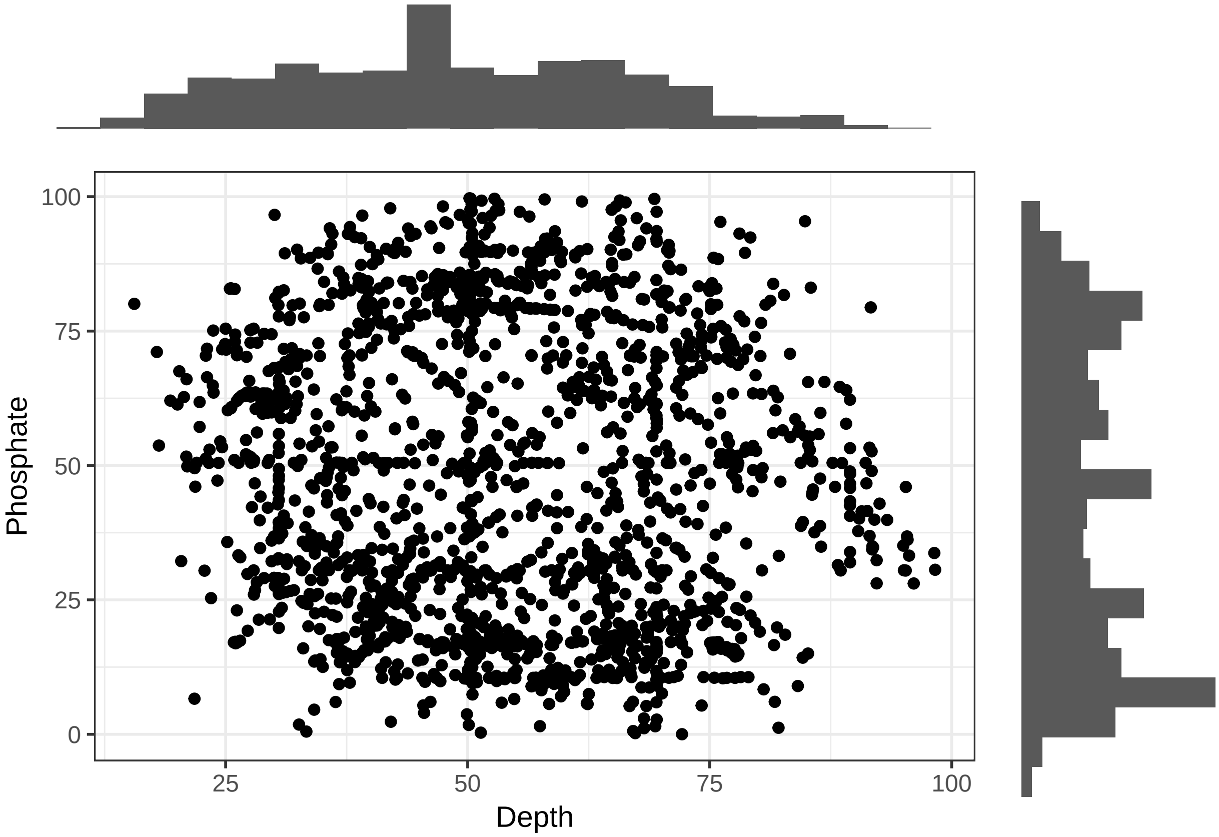
```
# not sure what else I need besides pairs plot
p1 <- ggplot(data = data) +
  geom_mosaic(aes(x = product(day, block), fill = day)) +
  labs(x = "Block", y = "Day") +
  ggtitle("Mosaic Plot of Block and Day") +
  theme_bw()

 p2 <- ggplot(data = data) +
   geom_density(aes(x = depth, fill = day), alpha = 0.5) +
   labs(x = "Depth", y = "Density") +
   ggtitle("Density of Depth by Day") +
   theme_bw()

p3 <- ggplot(data = data) +
   geom_density(aes(x = phosphate, fill = day), alpha = 0.5) +
   labs(x = "Phosphate", y = "Density") +
   ggtitle("Density of Phosphate by Day") +
   theme_bw()

p4 <- ggplot(data = data, aes(x = depth, y = phosphate)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x) +
  labs(x = "Depth", y = "Phosphate") +
  ggtitle("Phosphate vs Depth") +
  theme_bw()

multiplot(p1, p2, p3, p4, cols = 2)
```



The lesson is that sometimes numbers are factors.

Problem 5

```
p5 <- ggplot(data = data) +
  geom_point(aes(x = depth, y = phosphate)) +
  labs(x = "Depth", y = "Phosphate") +
  theme_bw()

p6 <- ggplot(data = data) +
  geom_histogram(aes(x = depth), bins = 20) +
  theme_classic() +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        line = element_blank())

p7 <- ggplot(data = data) +
  geom_histogram(aes(x = phosphate), bins = 20) +
  coord_flip() +
  theme_classic() +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        line = element_blank())

# histograms appear slightly shifted too low since I cut off the axis
grid.arrange(
  grobs = list(p5, p6, p7),
  layout_matrix = rbind(c(2, 2, 2, 2, NA),
                        c(1, 1, 1, 1, 3),
                        c(1, 1, 1, 1, 3),
                        c(1, 1, 1, 1, 3),
                        c(1, 1, 1, 1, 3)))
```



Problem 6

Done.