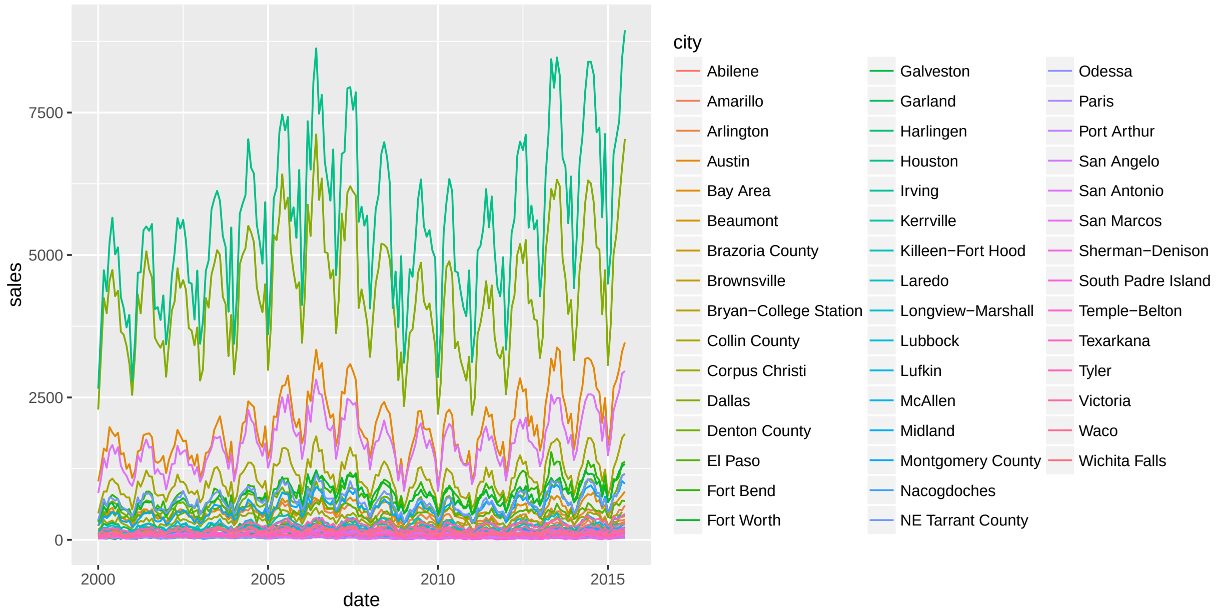
STAT 579 Homework 8

Yifan Zhu

November 14, 2016

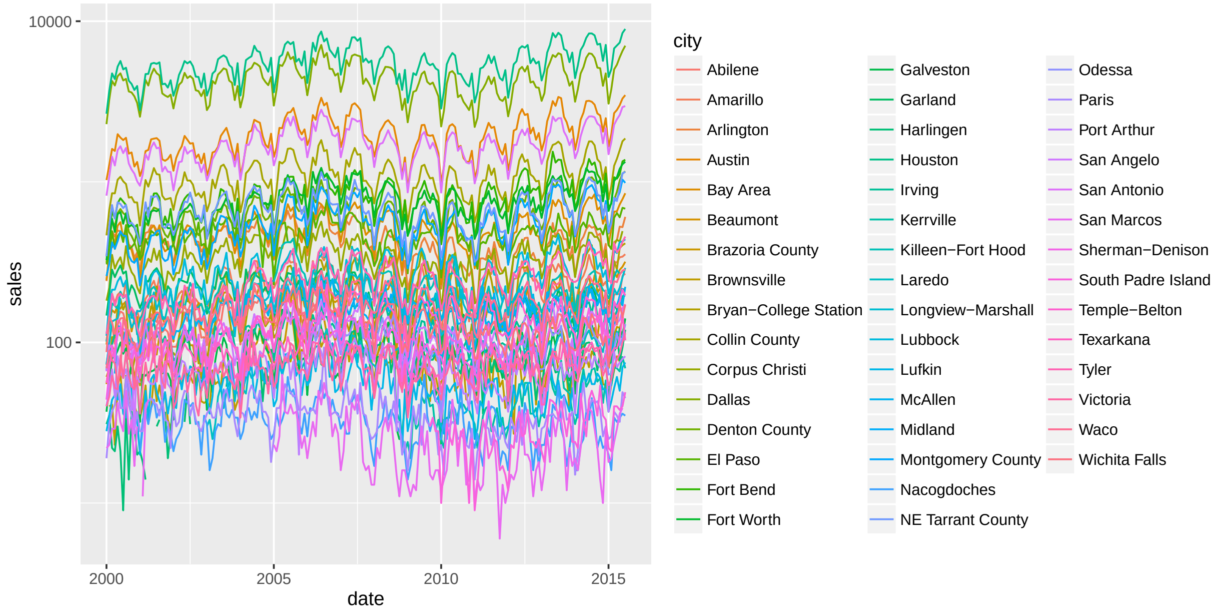
Problem 1 (a)

```
library(ggplot2)

salevsdata <- ggplot(data = txhousing, aes(x = date, y = sales, color = city)) +
    geom_line()

salevsdata
```



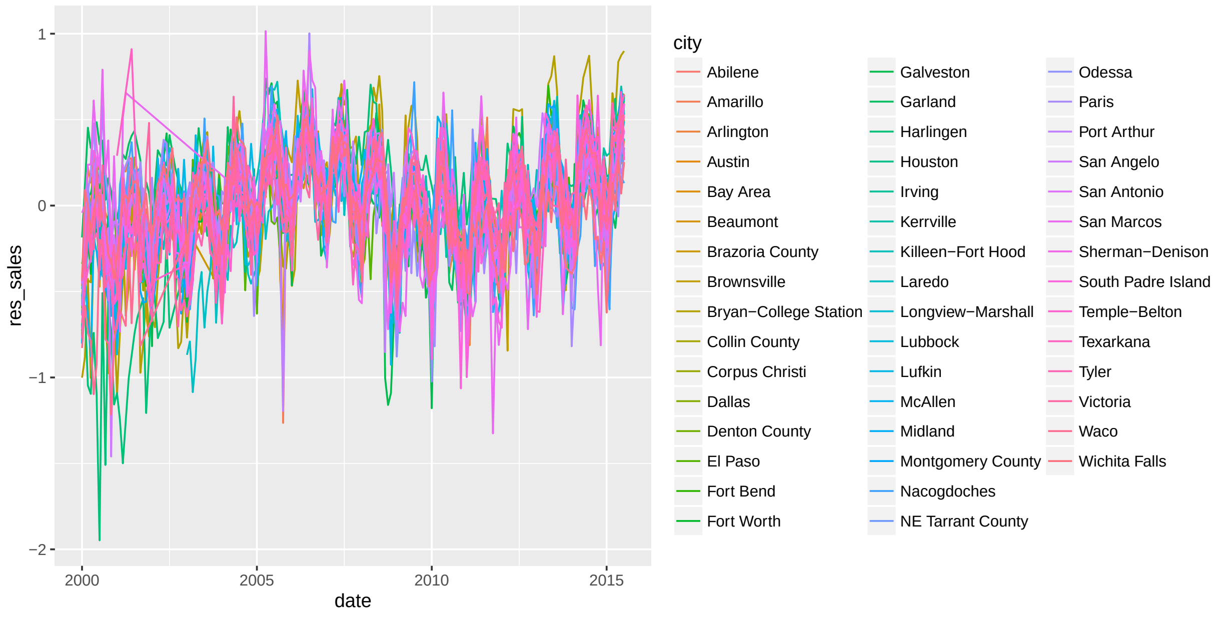
(b)

```
salevsdata + scale_y_log10()
```



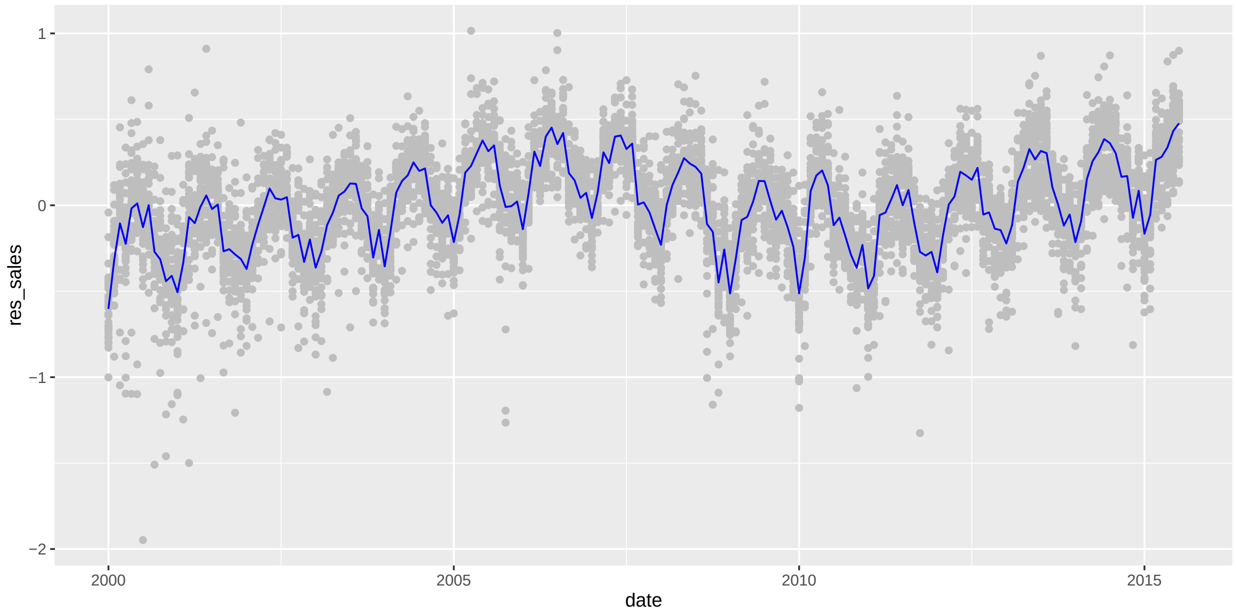
(c)

```
txhousing <- txhousing[!is.na(txhousing$sales), ]
txhousing$logsales <- log(txhousing$sales)
logsalemonthlm <- lm(formula = logsales ~ month + city + month:city, data = txhousing)
txhousing$res_sales <- logsalemonthlm$residuals

ggplot(data = txhousing, aes(x = date, y = res_sales, color = city)) + geom_line()
```



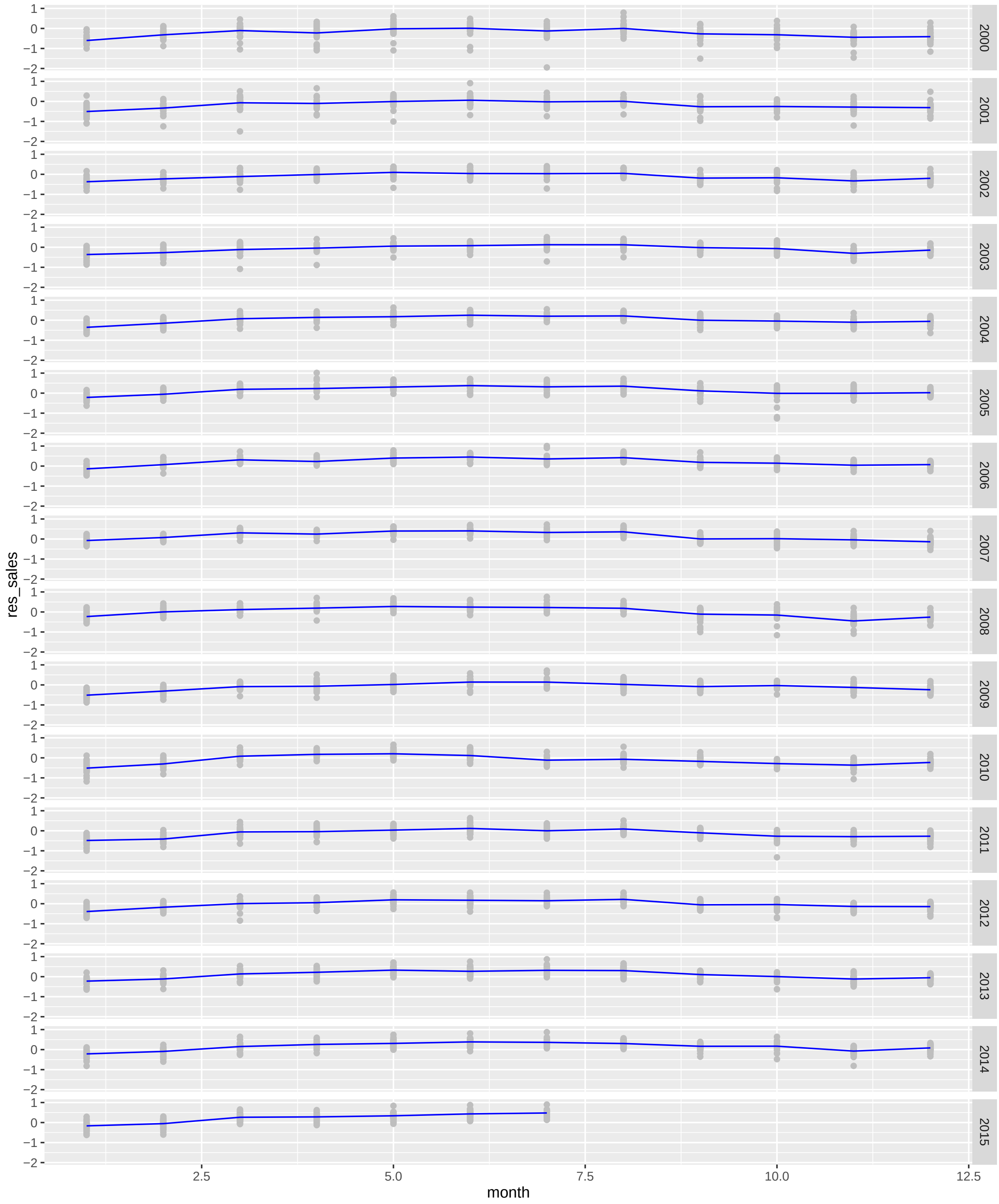
(d)

```r
ggplot(data = txhousing, aes(x = date, y = res_sales)) + geom_point(colour = "grey") +
    stat_summary(fun.y = mean, colour = "blue", geom = "line")
```



(e)

```
ggplot(data = txhousing, aes(x = month, y = res_sales)) + geom_point(colour = "grey") +
    stat_summary(fun.y = mean, colour = "blue", geom = "line") + facet_grid(year ~
    .)
```



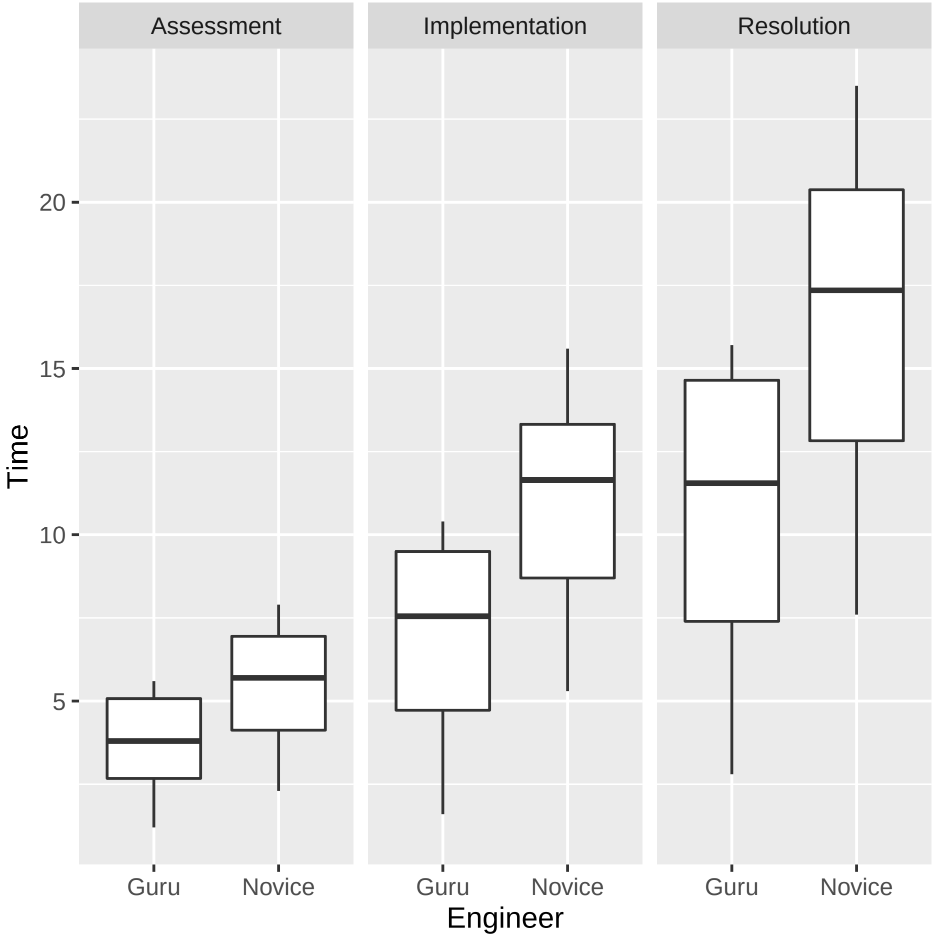
Problem 2

(a)

```
breakdown <- read.table(file = "http://maitra.public.iastate.edu/stat501/datasets/breakdown.dat")

names(breakdown) <- c("Problem", "Severity", "Engineer", "Assessment", "Implementation",
    "Resolution")
```

(b)

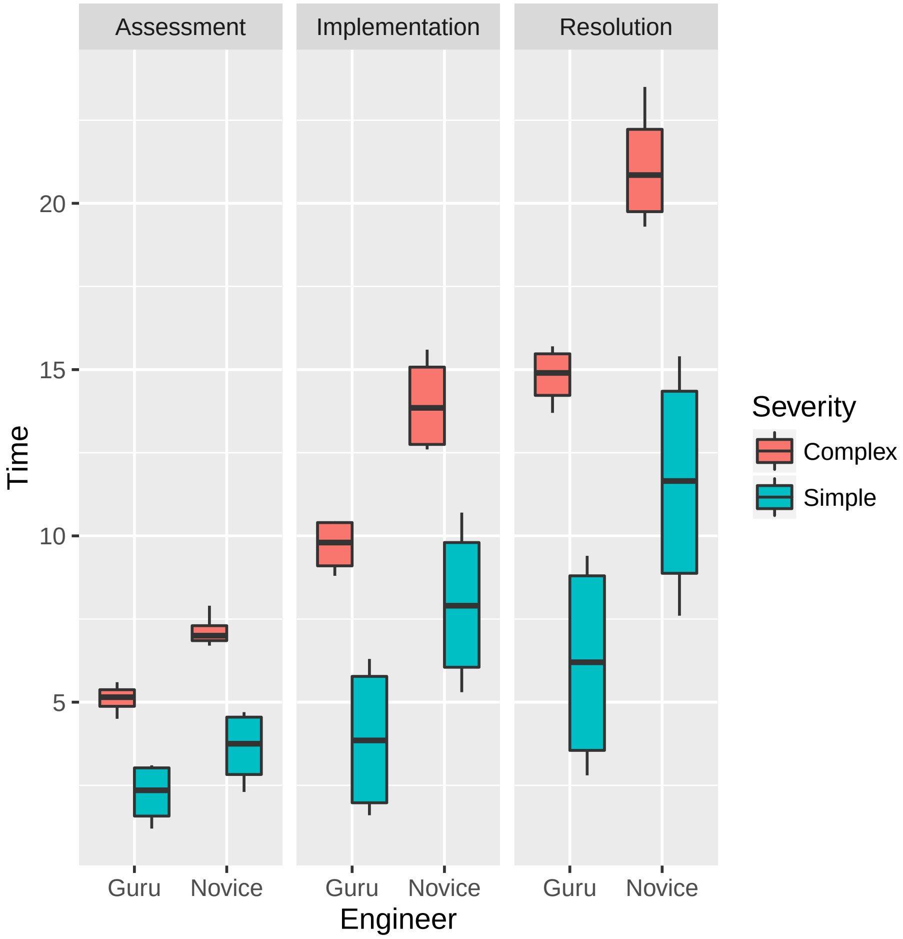
```
library(reshape2)

mbreakdown <- melt(breakdown, id.vars = c("Engineer", "Problem", "Severity"))

ggplot(data = mbreakdown, aes(x = Engineer, y = value)) + geom_boxplot() + facet_grid(. ~
    variable) + ylab("Time")
```



The mean time of assessment, implementation and resolution for guru engineer is shorter than that of novice engineer; the spread of resolution time is larger than spread of implemention, and spread of implemention time is larger than spread of assessment.

(c)

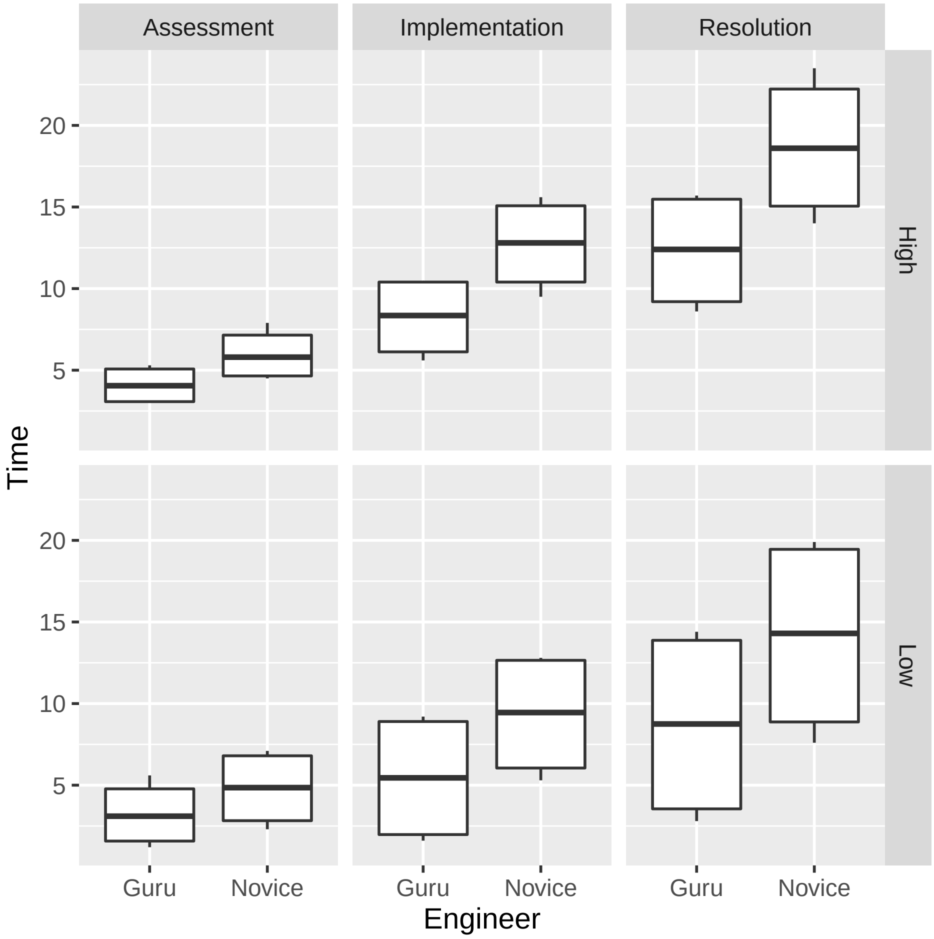
```
ggplot(data = mbreakdown, aes(x = Engineer, y = value, fill = Severity)) + geom_boxplot() +
    facet_grid(. ~ variable) + ylab("Time")
```



For each time of the three kinds, the mean time to handle the complex problem is larger than that of the simple problem for both guru and novice engineers. And the spread of the time for the complex problem is also larger than that of simple problem for guru engineer, but the spread of time for complex and simple problems are similar for novice engineer.

(d)

```
ggplot(data = mbreakdown, aes(x = Engineer, y = value)) + geom_boxplot() + facet_grid(Problem ~
    variable) + ylab("Time")
```



For problem whose severity is high, the ranges of time of three kinds for novice and guru almost have no overlap, which means that the guru engineer can always spend shorter time than guru engieer when fixing the problem with high severity. But for the problem with low severity, there is obvious overlap between the times for novice and guru engineers, so it is likely the the guru and novice spend the same time when fixing a problem with low severity.

Problem 3

(a)

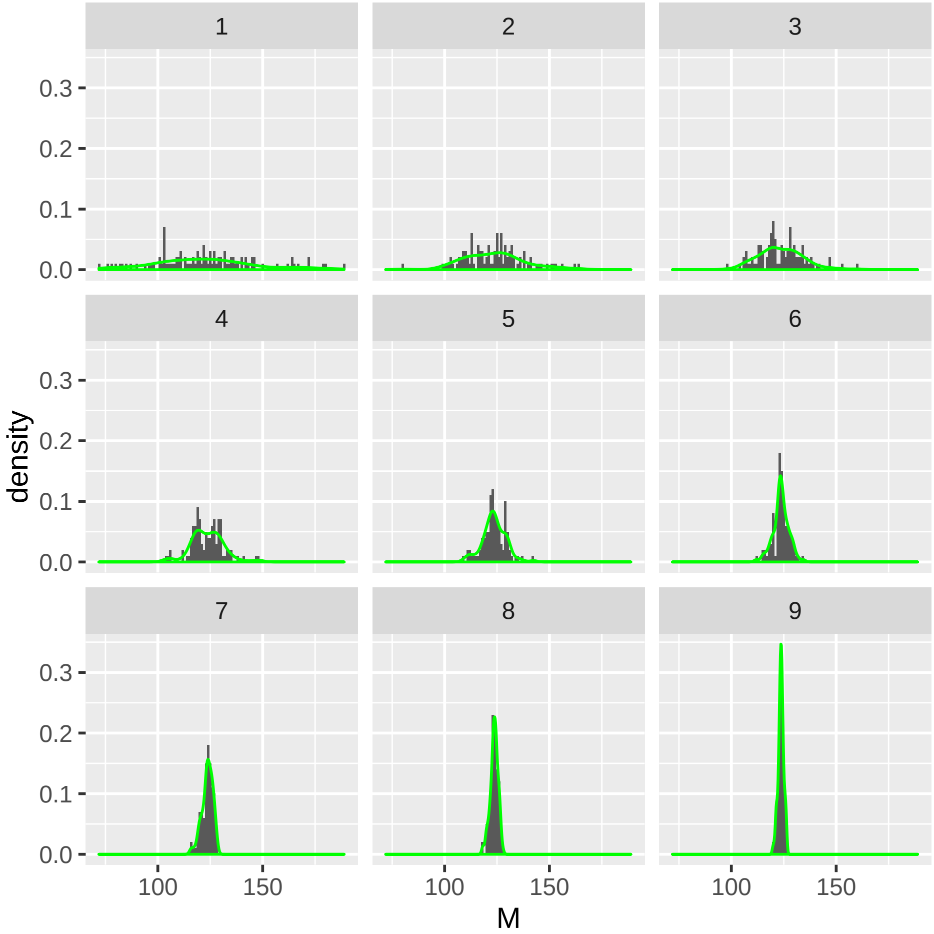
```
height <- data.frame(M = rnorm(mean = 125, sd = 25, n = 100), F = rnorm(mean = 125,
    sd = 15, n = 100))
```

(b)

```
height.ng <- function(height) {
    permutation <- sample(1:100, size = 100)
    height$M <- height$M[permutation]
    ngheight <- apply(height, MARGIN = 1, FUN = mean)
    return(data.frame(M = ngheight, F = ngheight))
}
```

(c)

```
heights9gen <- NULL
for (i in 1:9) {
    generation <- rep(i, 100)
    heights <- cbind(generation, height)
    heights9gen <- rbind(heights9gen, heights)
    height <- height.ng(height)
}

ggplot(data = heights9gen, aes(x = M, y = ..density..)) + geom_histogram(binwidth = 1) +
    geom_density(colour = "green") + facet_wrap(~generation, nrow = 3)
```



Problem 4

(a)

```
vectorization <- function(file) {
    out <- readLines(con = file)
    clusterid <- (1:length(out))[out == ""] + 1
    clusterstr <- out[clusterid]
    clusterstr <- clusterstr[!is.na(clusterstr)]
    sizes <- unlist(lapply(strsplit(clusterstr, split = "="), FUN = "[", 2))
    mode(sizes) <- "numeric"
    group <- NULL
    for (i in 1:length(sizes)) {
        group <- c(group, rep(i - 1, sizes[i]))
    }

    observation <- NULL
    for (i in 1:length(sizes)) {
        observation <- as.numeric(c(observation, out[(clusterid[i] + 1):(clusterid[i +
            1] - 2)]))
    }

    vec <- group[order(observation)]

    return(vec)
}
```

(b)

```
iris1file <- "C:/Users/fanne/Desktop/STAT579/STAT579hw8/Iris1.out"
iris2file <- "C:/Users/fanne/Desktop/STAT579/STAT579hw8/Iris2.out"

IrisGroup1 <- vectorization(iris1file)
IrisGroup2 <- vectorization(iris2file)

table(IrisGroup1, IrisGroup2)
```

```
##           IrisGroup2
## IrisGroup1  0  1
##          0 37  0
##          1  0 50
##          2 63  0
```