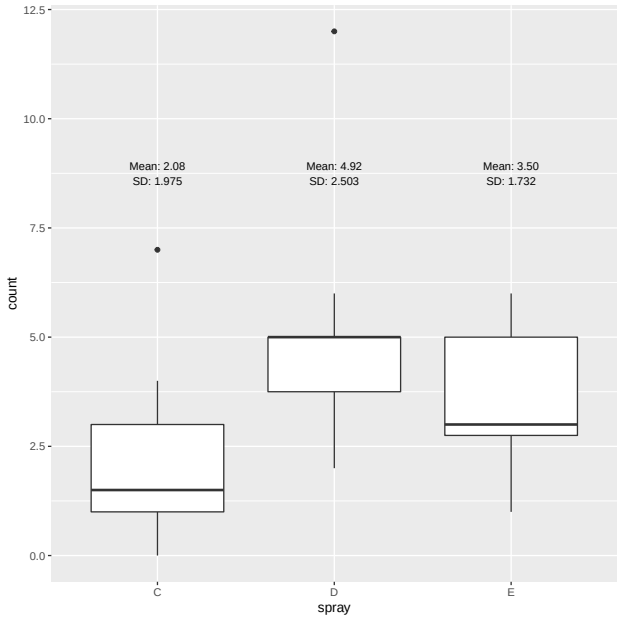
Stat 574B: HW 3 Problem 3

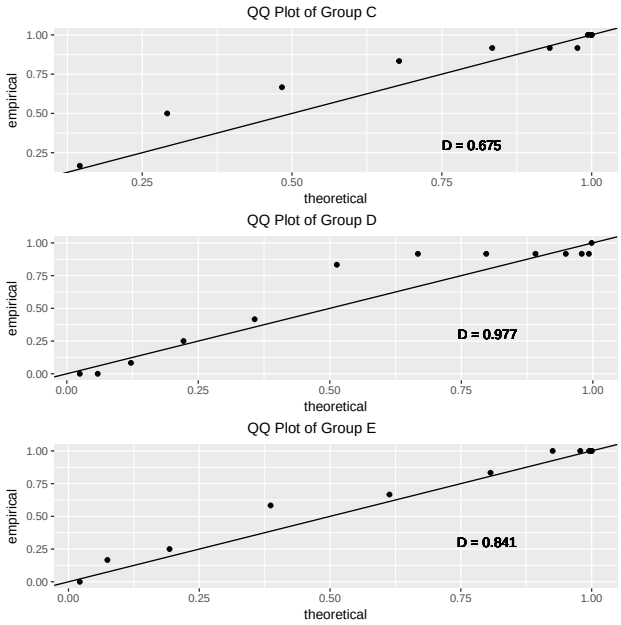
Nate Hattersley

November 1, 2017

part a.

I create box plots of the different groups, and then QQ plots by group. I include the D statistic from the Kolmogorov-Smirnov test against a normal distribution. I get a warning saying that ks.test doesn't like repeated values, so the results are not trustworthy. I took it as more of an exercise in labeling plots in ggplot...





Looking at the QQ plots, there do seem to be some serious issues with normality. All the groups are right-skewed. However, we're tasked with ANOVA analysis in this problem, hence the issue must not be too egregious.

part b.

I calculated means and confidence intervals from the posterior in the notes, given a reference prior. Stole the MLE estimate of σ^2 from the summary.lm object. I didn't do simulation, so I used the t distribution for the marginals of μ_i. Probability of being > 0 is included via the CDF of each reference distribution. For the contrasts, I used *combn* to pick every possible combination of two factors and *apply* to create my summary statistics of said contrast. Libraries dplyr and magrittr used to chain commands rather than nest them (I argue it produces much more comprehensible code).

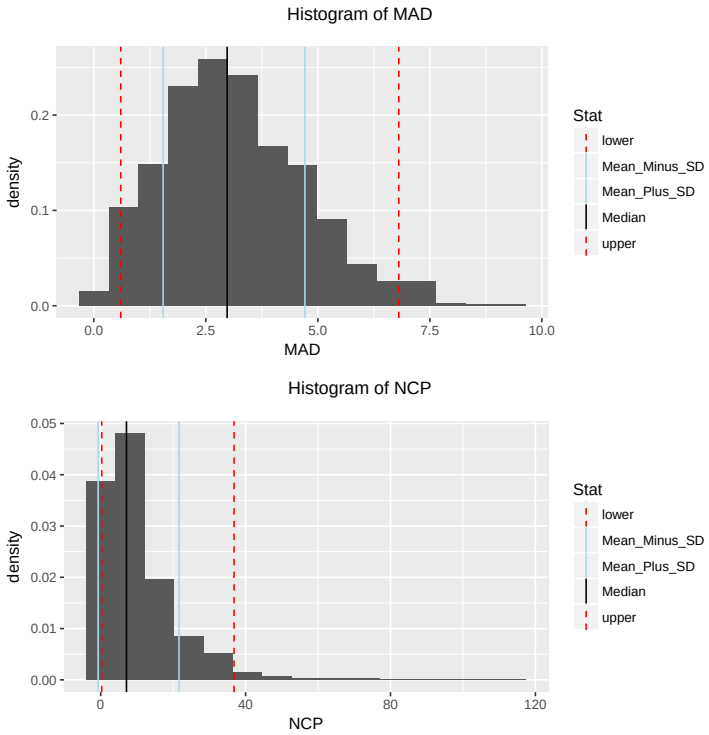
Effect	Mean	SD	CI	Pr.gt.0
C	2.0833	0.6048	0.85 to 3.31	0.9992
D	4.9167	0.6048	3.69 to 6.15	1.0000
E	3.5000	0.6048	2.27 to 4.73	1.0000
C-D	-2.8333	0.8553	-4.57 to -1.09	0.0011
C-E	-1.4167	0.8553	-3.16 to 0.32	0.0536
D-E	1.4167	0.8553	-0.32 to 3.16	0.9464

part c.

My simulation of MAD and NCP commenced thus: For each sample, I took three draws from a t distribution, $\mathbf{t}$, then computed mean estimates for each group by centering on the group mean and scaling by MSE * t_i. I then took the grand mean $\bar{\mu}$ to be the mean of these three draws, and σ^2 I drew randomly from the posterior in the notes. I went back and forth on what quantities to use as $\bar{\mu}, \sigma^2$. I erred on the side of more randomness.

I include histograms with the equal-tails 95% confidence interval on the statistic, as well as median and the range mean±SD. The distribution of MAD values appears to be closer to symmetric than the NCP does. This makes sense to me, as NCP has a $\sum_i (\mu_i - \bar{\mu})^2$ term that looks like a variance calculation, and we use the heavily skewed Inverse-Gamma distribution to sample variances.

	Mean	Median	SD	CI
MAD	3.13	2.98	1.58	0.602 to 6.807
NCP	10.46	7.12	11.16	0.284 to 36.813



part d.

To calculate the BIC for each model, I simply used window functions in dplyr to create density columns for the grouped and global models, then took their respective products for the likelihood. For DIC, I set $c = 0$ and used Monte Carlo methods to sample from $D(\theta)$. Instead of storing the output of the MC method in a matrix, I just kept a running average of deviance. Interestingly, BIC yielded a Bayes Factor of 10.31 in favor of choosing a model with just one mean (substantial to strong on Jeffery's scale), but DIC was lower for the model with group means. Here are the summaries of DIC and BIC for each model:

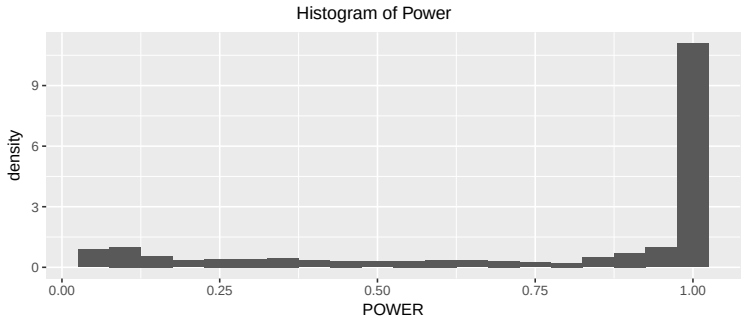
	Grouped	Global
DIC	160.21	166.68
BIC	-190.93	-186.26

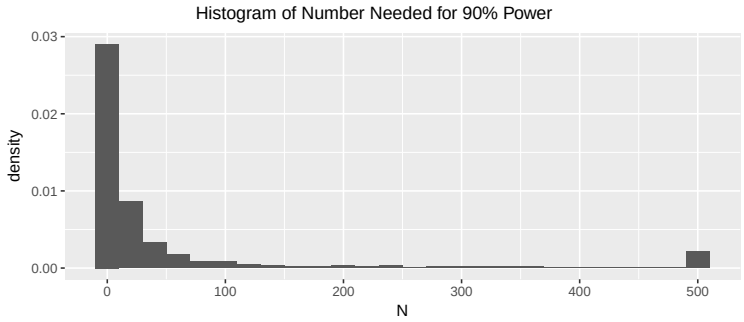
part e.

Since we've already calculated the non-centrality parameter ζ_1 from here, let's do some power analysis. I originally calculated the power to detect any difference in means, but that was too strong and I opted for a more interesting contrast. I choose $C - D + E = 0$, since, very roughly, C is 2, D is 5, and E is 3. For each draw from my MC sample in part c, I calculated the non-centrality parameter for the contrast. Then, fixing $\alpha = .05$, I took the area above the F critical point as power.

I also wanted to figure out the number of observations needed per group to attain 90% power. For each row in output, I looped over n until power was .9 or I hit an arbitrary cutoff of 500. (I figure these last ones are anomalous samples that don't warrant the processing power required to find the true n.)

I include histograms of each statistic, and medians since the distributions are heavily skewed.





Stat	Median
Power	0.99215
NN90P	7.00000

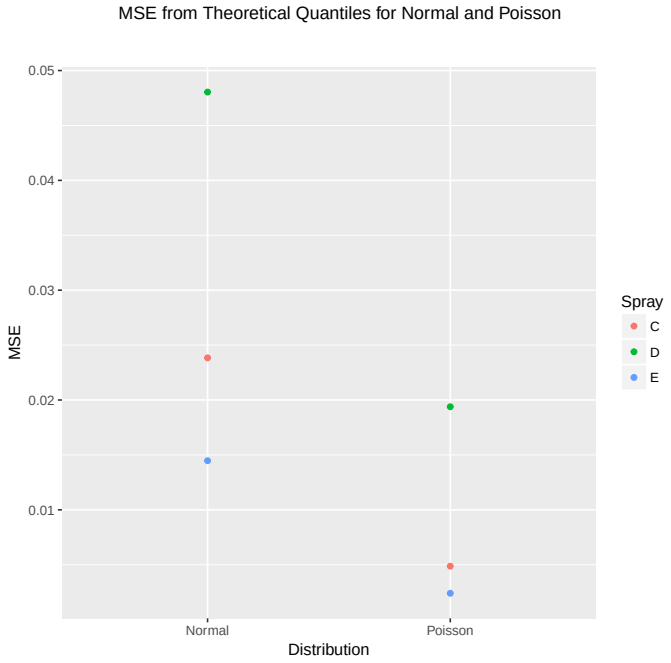
To give perspective, I found that 51.3% of the power samples were greater than 0.99. I also found that the probability of needing more than the original number of samples to get a sufficient power was 0.382. The percentage of samples that reached my cutoff of 500 was 4.4%.

Extra Credit

I threw several things at the wall in an attempt to make something stick. First, I tried Box's p-value method with a different test statistic for both Poisson and Normal: the VMR and skewness, respectively. Although this seems flawed to me since we are measuring different characteristics for each distribution, reference the wall and sticking comment above. In this analysis, I found a p value of 7.6293945×10^{-6} for the probability that the population is Poisson-distributed, and a p-value of 4.8828125×10^{-4} for the probability that it is Normal.

Then, after you made your comment in class on Monday about adjusting the normal distribution, I thought of doing BIC to compare the two models. I would have to integrate the Normal distribution to have a bin width of 1 like Poisson. I chose to take a Riemann sum from $x - 0.5$ to $x + 0.5$ for each integer x to approximate the probability under a unit width of input. Using the altered definition of *dnorm*, I found a Bayes Factor of 3.432 in favor of the Poisson model.

Finally, I wanted to get creative and do something we hadn't been taught. I regressed the empirical quantiles against the theoretical for both the Poisson and Normal, and calculated the MSE as a measure of discrepancy. I did this under the assumption of non-identically distributed groups, so there are three data points for each hypothesized distribution. At a glance, it looks like Poisson more closely approximates the group distributions.



Code

```
### PART A
data("InsectSprays")
data.df <- InsectSprays %>% filter(spray %in% c("C", "D", "E"))

boxplot.df <- data.df %>% group_by(spray) %>% summarise(mean = mean(count),
    sd = sd(count))


boxplot.df %<>% mutate(lab = sprintf("Mean: %.2f\nSD: %.3f",
    mean, sd))
ggplot(data.df, aes(x = spray, y = count)) + geom_boxplot() +
    geom_text(data = boxplot.df, aes(x = spray, y = 8.75, label = lab,
        hjust = 0.5), size = 3)

n <- length(data.df[, 1])

plots = list()

levs <- data.df %$% spray %>% factor %>% levels

for (i in levs) {
    subset.vec <- data.df %>% filter(spray == i) %>% extract2(1)

    qqplot.df <- data.frame(empirical = ecdf(subset.vec)(0:12),
        theoretical = pnorm(0:12, mean(subset.vec), sd(subset.vec)))
    test <- invisible(suppressWarnings(ks.test(subset.vec, "pnorm")))
    plots[[i]] <- arrangeGrob(ggplot(qqplot.df) + geom_point(aes(theoretical,
        empirical)) + geom_abline(slope = 1, intercept = 0) +
        geom_text(x = 0.8, y = 0.3, label = sprintf("D = %.3f",
            test$statistic)), top = sprintf("QQ Plot of Group %s",
        i))

}

do.call(grid.arrange, plots)
#### PART B Posterior analysis
XTX <- data.df %>% count(spray) %$% n %>% sapply(function(n) {
    1/n
}) %>% diag
MU <- boxplot.df %$% mean %>% set_names(c("C", "D", "E")) %>%
    matrix(ncol = 1)
S <- data.df %>% lm(count ~ spray, data = .) %>% summary %$%
    sigma %>% raise_to_power(2)



summary.df <- data.frame(Effect = character(), Mean = numeric(),
    SD = numeric(), CI = character(), `Pr gt 0` = numeric())
```

```
talpha <- qt(0.975, n - 3)
for (i in 1:length(levs)) {
    s <- sqrt(S * diag(XTX)[i])
    summary.df %<>% rbind(data.frame(Effect = as.character(levs[i]),
        Mean = MU[i, 1], SD = s, CI = sprintf("%.2f to %.2f",
            MU[i, 1] - s * talpha, MU[i, 1] + s * talpha), `Pr gt 0` = 1 -
            pt(-1 * MU[i]/s, n - 3)))

}
combn(levs, 2) %>% apply(2, function(factors) {
    i1 <- (levs == factors[1]) %>% as.numeric
    i2 <- (levs == factors[2]) %>% as.numeric %>% multiply_by(-1)
    C <- (i1 + i2) %>% matrix(ncol = 1)
    mu <- (t(C) %*% MU) %>% as.numeric
    s <- (sqrt(S * (t(C) %*% XTX %*% C))) %>% as.numeric
    summary.df <<- rbind(summary.df, data.frame(Effect = sprintf("%s-%s",
        factors[1], factors[2]), Mean = mu, SD = s, CI = sprintf("%.2f to %.2f",
        mu - s * talpha, mu + s * talpha), `Pr gt 0` = 1 - pt(-1 *
        mu/s, n - 3)))
}) %>% invisible

print(xtable(summary.df, digits = 4), include.rownames = F)

### PART C
set.seed(12345)
mcmcSize <- 1000

output <- matrix(ncol = 5, nrow = mcmcSize) %>% set_colnames(c("C",
    "D", "E", "MAD", "NCP"))

for (i in 1:mcmcSize) {
    t <- rt(3, n - 3)
    estimate <- t * S + c(MU)
    MUbar <- mean(estimate)
    S.estimate <- 1/rgamma(1, 0.5 * (n - 3), 0.5 * (n - 3) *
        S)
    output[i, ] <- c(estimate, sum(abs(estimate - MUbar)/3),
        sum((estimate - MUbar)^2/S.estimate))
}

output %<>% as.data.frame


stat.summary <- output[, 4:5] %>% tidyr::gather("Statistic") %>%
    group_by(Statistic) %>% summarise(Mean = mean(value), Median = median(value),
    SD = sd(value), lower = quantile(value, 0.025), upper = quantile(value,
        0.975)) %>% as.data.frame

rownames(stat.summary) <- stat.summary[, 1]
stat.summary <- stat.summary[, -1]
```

```
table.df <- stat.summary %>% mutate(CI = sprintf("%.3f to %.3f",
    lower, upper)) %>% select(Mean, Median, SD, CI) %>% set_rownames(rownames(stat.summary))

print(xtable(table.df))

stat.plotdf <- stat.summary %>% mutate(Mean_Minus_SD = Mean -
    SD, Mean_Plus_SD = Mean + SD) %>% select(-SD, -Mean) %>%
    t %>% as.data.frame %>% set_colnames(c("MAD", "NCP"))

stat.plotdf %<>% cbind(Stat = rownames(stat.plotdf))


MADPLOT <- (ggplot(output) + geom_histogram(aes(x = MAD, y = ..density..),
    bins = 15) + geom_vline(aes(xintercept = MAD, col = Stat,
    linetype = Stat), data = stat.plotdf) + scale_color_manual(values = c("red",
    "lightblue", "lightblue", "black", "red")) + scale_linetype_manual(values = c("dashed",
    "solid", "solid", "solid", "dashed")) + theme(plot.margin = unit(c(rep(1,
    4)), "lines"))) %>% arrangeGrob(top = "Histogram of MAD")

NCPPLOT <- (ggplot(output) + geom_histogram(aes(x = NCP, y = ..density..),
    bins = 15) + geom_vline(aes(xintercept = NCP, col = Stat,
    linetype = Stat), data = stat.plotdf) + scale_color_manual(values = c("red",
    "lightblue", "lightblue", "black", "red")) + scale_linetype_manual(values = c("dashed",
    "solid", "solid", "solid", "dashed")) + theme(plot.margin = unit(c(rep(1,
    4)), "lines"))) %>% arrangeGrob(top = "Histogram of NCP")

grid.arrange(MADPLOT, NCPPLOT)


### PART D Model selection

BIC.df <- data.df %>% mutate(mu = mean(count)) %>% group_by(spray) %>%
    mutate(mu_i = mean(count)) %>% mutate(dens = dnorm(count,
    mu, S), dens_i = dnorm(count, mu_i, S))

BIC.stat <- with(BIC.df, c(Grouped = 2 * (dens_i %>% prod %>%
    log) - 3 * log(n), Global = 2 * (dens %>% prod %>% log) -
    log(n)))


## evidence in favor of single-mean model
BF <- exp(0.5 * (BIC.stat["Global"] - BIC.stat["Grouped"]))



L <- function(mu, sigma.sq, model = NA, y = data.df) {
    if (is.na(model) || !model %in% c("grouped", "global"))
        stop("Specify a model")

    y %>% apply(1, function(r) {
```

```
        theta <- ifelse(model == "grouped", mu[r["spray"]], mu)
        x <- r["count"] %>% as.numeric
        dnorm(x, theta, sd = sqrt(sigma.sq))
    }) %>% prod
}
D <- function(...) {
    -2 * log(L(...))
}


Shat <- var(data.df$count)
MU.global <- mean(data.df$count)

## Sample from posterior of both models and keep a running
## average of D
mcmcSize <- 1000
ED <- c(0, 0)  ## Pun not intended
for (i in 1:mcmcSize) {
    sigma.global <- 1/rgamma(1, 0.5 * (n - 1), 0.5 * (n - 1) *
        Shat)
    mu.global <- rnorm(1, MU.global, sqrt(sigma.global/n))
    sigma.grouped <- 1/rgamma(1, 0.5 * (n - 3), 0.5 * (n - 3) *
        S)
    mu.grouped <- mvrnorm(1, MU, sigma.grouped * XTX) %>% set_names(c("C",
        "D", "E"))

    ED[1] %<>% c(D(mu.grouped, sigma.grouped, "grouped")) %>%
        weighted.mean(c(i - 1, 1))
    ED[2] %<>% c(D(mu.global, sigma.global, "global")) %>% weighted.mean(c(i -
        1, 1))
}



ES.global <- (0.5 * (n - 1) * Shat)/(0.5 * (n - 1) - 1)
ES.grouped <- (0.5 * (n - 1) * S)/(0.5 * (n - 1) - 1)

MU.grouped <- MU %>% c %>% set_names(c("C", "D", "E"))

Dhat <- c(D(MU.grouped, ES.grouped, "grouped"), D(MU.global,
    ES.global, "global"))

DIC <- (2 * ED - Dhat) %>% matrix(nrow = 1) %>% set_colnames(c("Grouped",
    "Global")) %>% set_rownames(c("DIC"))

### PART E
contrast <- c(1, -1, 1)

NCP <- output %>% apply(1, function(r) {
    means <- c(r[1], r[2], r[3])
    sum(contrast * means)^2/(S * sum(contrast^2/12))
```

```
})

fcrit <- qf(0.95, 2, 33)

POWER <- 1 - pf(fcrit, 2, 33, NCP)


POW.n <- function(means, n) {
    NCP.n <- sum(contrast * means)^2/(S * sum(contrast^2/n))
    1 - pf(fcrit, 2, 33, NCP.n)
}

POWER.n <- output %>% apply(1, function(r) {
    means <- c(r[1], r[2], r[3])
    i <- 1
    while (POW.n(means, i) < 0.9 && i < 500) {
        i <- i + 1
    }
    i
})

POWER.plot <- arrangeGrob(ggplot(data.frame(POWER = POWER)) +
    geom_histogram(aes(POWER, ..density..), binwidth = 0.05),
    top = "\n\nHistogram of Power")

POWER.n.plot <- arrangeGrob(ggplot(data.frame(N = POWER.n)) +
    geom_histogram(aes(N, ..density..), binwidth = 20), top = "\n\nHistogram of Number Needed for 90% Po

grid.arrange(POWER.plot, POWER.n.plot)

data.frame(Stat = c("Power", "NN90P"), Median = sapply(list(POWER,
    POWER.n), median)) %>% xtable(digits = 5) %>% print(include.rownames = F)

## EXTRA CREDIT Box's p-value analysis Assuming one group, not
## three, for simplicity Using VMR as T for Poisson, skew for
## Normal

VMR <- function(i) {
    var(i)/mean(i)
}
skewness <- function(i) {
    mean((i - mean(i))^3)/(var(i)^1.5)
}

Tobs.norm <- skewness(data.df$count)
Tobs.pois <- VMR(data.df$count)

mcmcSize <- 1000

pval.pois <- numeric(0)
pval.norm <- numeric(0)
```

```
pval.update <- function(p, ..., model) {
    samp <- switch(model, normal = abs(skewness(rnorm(n, ...))) >
        abs(Tobs.norm), poisson = VMR(rpois(n, ...)) > Tobs.pois,
        stop("No model supplied to pval.update"))

    ifelse(length(p) == 0, as.numeric(samp), c(p, samp) %>% weighted.mean(c(length(p),
        1)))
}

for (i in 1:mcmcSize) {
    theta.pois <- rgamma(1, sum(data.df$count), n)
    sigma.norm <- (1/rgamma(1, 0.5 * (n - 1), 0.5 * (n - 1) *
        Shat)) %>% sqrt
    theta.norm <- rnorm(1, mean(data.df$count), sigma.norm)
    pval.pois %<>% pval.update(theta.pois, model = "poisson")
    pval.norm %<>% pval.update(theta.norm, sigma.norm, model = "normal")

}

## revised dnorm function to calculate area from x-0.5 to
## x+0.5
ddnorm <- function(n) {
    sapply(n, function(x) {
        delta <- 0.01
        mu <- mean(data.df$count)
        s <- sd(data.df$count)
        seq(x - 0.5, x + 0.5, delta) %>% dnorm(mu, s) %>% sum %>%
            multiply_by(delta)
    })
}


BIC.df <- data.df %>% mutate(dens.pois = dpois(count, mean(count)),
    dens = ddnorm(count))

BIC.stat <- with(BIC.df, c(Poisson = 2 * (dens.pois %>% prod %>%
    log) - log(n), Normal = 2 * (dens %>% prod %>% log) - log(n)))


## This part assumes different means.  For each group,
## regresses the ECDF Against the CDF of the MLE model


MSE.qq <- list(Normal = numeric(), Poisson = numeric(), Spray = character())

for (i in levs) {
    subset.vec <- data.df %>% filter(spray == i) %>% extract2(1)

    qqtest.df <- data.frame((function(r) {
        list(empirical = ecdf(r)(r), normal = pnorm(r, mean(r),
```

```
            sqrt(S)), poisson = ppois(r, mean(r)))
    })(subset.vec))

    qqtest.df %<>% mutate(MSEnorm = (normal - empirical)^2, MSEpois = (poisson -
        empirical)^2)

    MSE.qq$Normal[i] <- sum(qqtest.df$MSEnorm)/length(subset.vec)
    MSE.qq$Poisson[i] <- sum(qqtest.df$MSEpois)/length(subset.vec)
    MSE.qq$Spray[i] <- i
}

graphing.df <- MSE.qq %>% as.data.frame %>% tidyr::gather("Distribution",
    "MSE", 1:2)

grid.arrange(ggplot(graphing.df, aes(Distribution, MSE, col = Spray)) +
    geom_point() + theme(plot.margin = unit(c(rep(2, 4)), "lines")),
    top = "\n\nMSE from Theoretical Quantiles for Normal and Poisson")
```