Gina Nichols - meta analysis of treatment diffs

Miranda Tilton 2021-01-29

1/28/2021

```
dat_miranda <- read_csv("./dat_miranda.csv")
```

```
ylds <-
  dat_miranda %>%
  select(site, year, rotation, nrate_kgha, yield_kgha) %>%
  pivot_wider(names_from = rotation, values_from = yield_kgha) %>%
  filter(!is.na(sc))

ylds
```

```
## # A tibble: 109 x 5
##    site   year nrate_kgha     cc     sc
##    <chr> <dbl>      <dbl>  <dbl>  <dbl>
##  1 ames   2000        270.  8157.  8725.
##  2 ames   2001        270.  6847.  9011.
##  3 ames   2002        270.  9510. 10878.
##  4 ames   2003        270.  7929. 10120.
##  5 ames   2004        270. 12336. 13076.
##  6 ames   2005        270.  8564. 10634.
##  7 ames   2006        270.  9749. 11573.
##  8 ames   2007        270. 10225. 10535.
##  9 ames   2008        270. 12046. 13142.
## 10 ames   2009        270. 10921. 11680.
## # ... with 99 more rows
```

```
sds <-
  dat_miranda %>%
  select(site, year, rotation, nrate_kgha, sd_kgha) %>%
  pivot_wider(names_from = rotation, values_from = sd_kgha) %>%
  rename("cc_sd" = cc,
         "sc_sd" = sc) %>%
  filter(!is.na(sc_sd))
sds
```

```
## # A tibble: 109 x 5
##    site   year nrate_kgha cc_sd sc_sd
##    <chr> <dbl>      <dbl> <dbl> <dbl>
##  1 ames   2000        270.  565.  573.
##  2 ames   2001        270.  452.  467.
##  3 ames   2002        270.  124.  606.
```

```
##  4 ames   2003       270.  487.  589.
##  5 ames   2004       270.  674.  605.
##  6 ames   2005       270.  521.  450.
##  7 ames   2006       270.  717.  533.
##  8 ames   2007       270.  174.  890.
##  9 ames   2008       270.  723.  214.
## 10 ames   2009       270.  312.  558.
## # ... with 99 more rows
```

```
dat <-
  ylds %>%
  left_join(sds) %>%
  mutate(gap_kgha = sc - cc)
```

```
## Joining, by = c("site", "year", "nrate_kgha")
```

```
dat
```

```
## # A tibble: 109 x 8
##    site   year nrate_kgha     cc     sc cc_sd sc_sd gap_kgha
##    <chr> <dbl>      <dbl>  <dbl>  <dbl> <dbl> <dbl>    <dbl>
##  1 ames   2000       270.  8157.  8725.  565.  573.     568.
##  2 ames   2001       270.  6847.  9011.  452.  467.    2164.
##  3 ames   2002       270.  9510. 10878.  124.  606.    1368.
##  4 ames   2003       270.  7929. 10120.  487.  589.    2190.
##  5 ames   2004       270. 12336. 13076.  674.  605.     740.
##  6 ames   2005       270.  8564. 10634.  521.  450.    2071.
##  7 ames   2006       270.  9749. 11573.  717.  533.    1824.
##  8 ames   2007       270. 10225. 10535.  174.  890.     310.
##  9 ames   2008       270. 12046. 13142.  723.  214.    1096.
## 10 ames   2009       270. 10921. 11680.  312.  558.     759.
## # ... with 99 more rows
```

1. How to estimate the uncertainty around the DIFFERENCE of cc and sc

The theory behind t-tests

For simpler notation, assume that we first choose a specific site and year. Then we have two populations represented by the two treatment groups (e.g., 1 = corn, 2 = soybean), where each population is normally distributed with its own mean but shared variance:

$$X_{1,i} \sim N(\mu_1, var = \sigma^2), \quad X_{2,i} \sim N(\mu_2, var = \sigma^2)$$

Then, given four reps ($i = 1, 2, 3, 4$), $\overline{X}_{1,\cdot} = \frac{1}{4}\sum_{i=1}^{4} X_{1,i}$ is the mean of those four reps for treatment 1 (corn), and is normally distributed

$$\overline{X}_{1\cdot} \sim N(\mu_1, var = \sigma^2/4)$$

The same holds for $\overline{X}_{2,\cdot}$. Then, assuming the two means are independent of each other,

$$\overline{X}_{1,\cdot} - \overline{X}_{2,\cdot} \sim N(\mu_1 - \mu_2, var = \frac{\sigma^2}{4} + \frac{\sigma^2}{4} = \frac{\sigma^2}{2})$$

$$\overline{X}_{1,\cdot} - \overline{X}_{2,\cdot} \sim N(\mu_1 - \mu_2, sd = \frac{\sigma}{\sqrt{2}})$$

The formulas

We assume that the two observed standard deviations σ_1, σ_2 are unbiased estimators of the true σ, so we can use the usual formula for pooling variance estimates (where s_1 and s_2 are the standard deviations provided in your data).

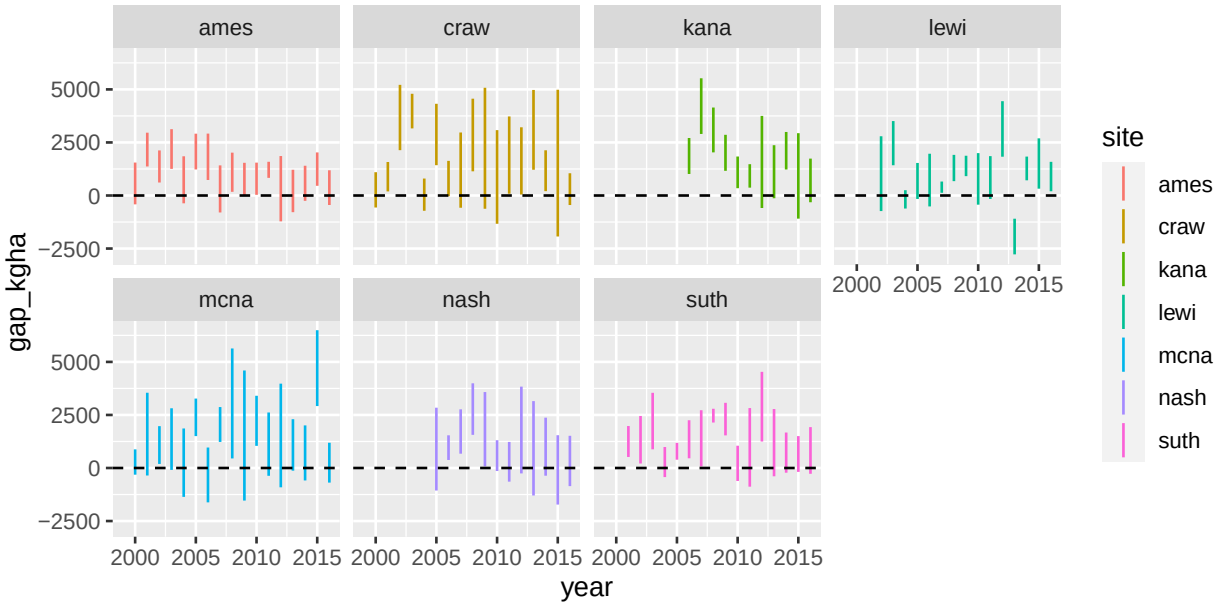
$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}$$

So the 95% CI is

$$\overline{X}_{1,\cdot} - \overline{X}_{2,\cdot} \pm t_{1-\alpha/2,df} * s_p * \sqrt{\frac{1}{n_1} + \frac{1}{n_2}},$$

where s_p is the unbiased estimator of σ and the final term simplifies to $\frac{1}{\sqrt{2}}$. We use a t critical value because we are estimating standard deviations/variances from a small number of cases.

```
crit_val <- qt(p = .975, df = 4 + 4 - 2) # p = 1 - (.05)/2

dat2 <- dat %>%
  mutate(cc_s2 = (cc_sd) ^ 2,
         sc_s2 = (sc_sd) ^ 2,
         sp = sqrt(3 * (cc_s2 + sc_s2) / (4 + 4 - 2)), # pool sp by year and site
         gap_se = sp * sqrt(1/4 + 1/4)) # standard error

dat2 %>%
  ggplot(aes(year, gap_kgha)) +
  geom_linerange(aes(ymin = gap_kgha - crit_val*gap_se,
                     ymax = gap_kgha + crit_val*gap_se,
                     color = site)) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  facet_wrap(~site, nrow = 2) # changed to facet_wrap, personal preference :)
```



The previous plot pooled the standard error information for each *pair* of treatments within a single site and year. If we believe that the variance is going to be similar for, say, all years within a site, then we can pool error information within a site using a similar procedure. Pooling variances can give us narrower confidence intervals.

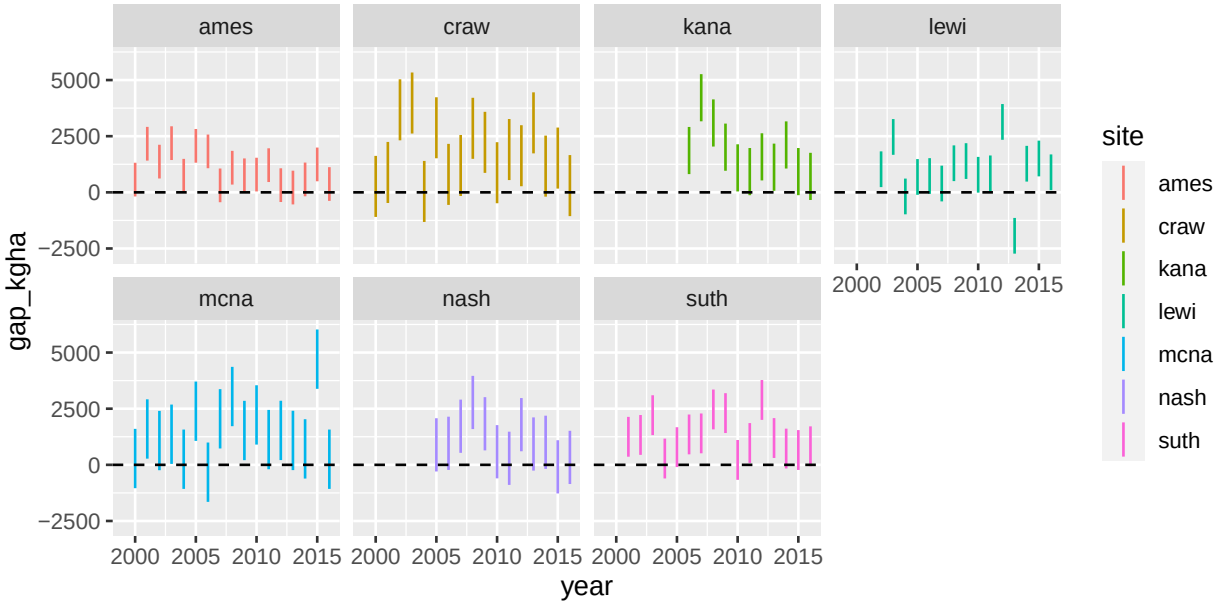
Essentially, to get s_p you square the stdev terms and take their average. It would be more complicated to weight them if they didn't all come from 4 reps, but in this balanced case it simplifies to just averaging them. The only piece then is the degrees of freedom, which is $4 * n$ where n is the number of *groups* you're averaging over, which is two (corn, soybean) per year per site. Recall that when $df > 30$, which is the case for all years within each site, the t-quantile can be a z-quantile from `qnorm()`.

```
table(dat2$site)
```

```
##
## ames craw kana lewi mcna nash suth
##   17   18   12   16   18   12   16
```

Since each site has more than 4 years, the site-wise pooled error will have more than $4*4*2 = 32$ df, meaning we can just use the z-quantile from `qnorm()`.

```
crit_val <- qnorm(p = .975)

dat3 <- dat2 %>%
  group_by(site) %>%
  mutate(sp_site = sqrt(mean(c(cc_s2, sc_s2), na.rm = TRUE)),
         gap_se_site = sp_site * sqrt(1/4 + 1/4)) # standard error


ggplot(dat3, aes(x = year, y = gap_kgha)) + # reuse same plot code from above
  geom_linerange(aes(ymin = gap_kgha - crit_val*gap_se_site,
                     ymax = gap_kgha + crit_val*gap_se_site,
                     color = site)) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  facet_wrap(~site, nrow = 2) # changed to facet_wrap, personal preference :)
```



2. You can assess significance visually, as you suggested, by determining whether the 95% CI contains zero. However, this is likely to produce false positives, since these are essentially 105 independent t-tests with false-positive probability $\alpha = 0.05$ for each test, so you can expect ≈ 5 false positive tests in the bunch.

ANOVA post-hoc testing uses Tukey's HSD to adjust all pairwise differences within a group, but you usually need all of the data to run these tests automatically. You can run them by hand, but you only care about the pairwise adjustments instead of family-wise.

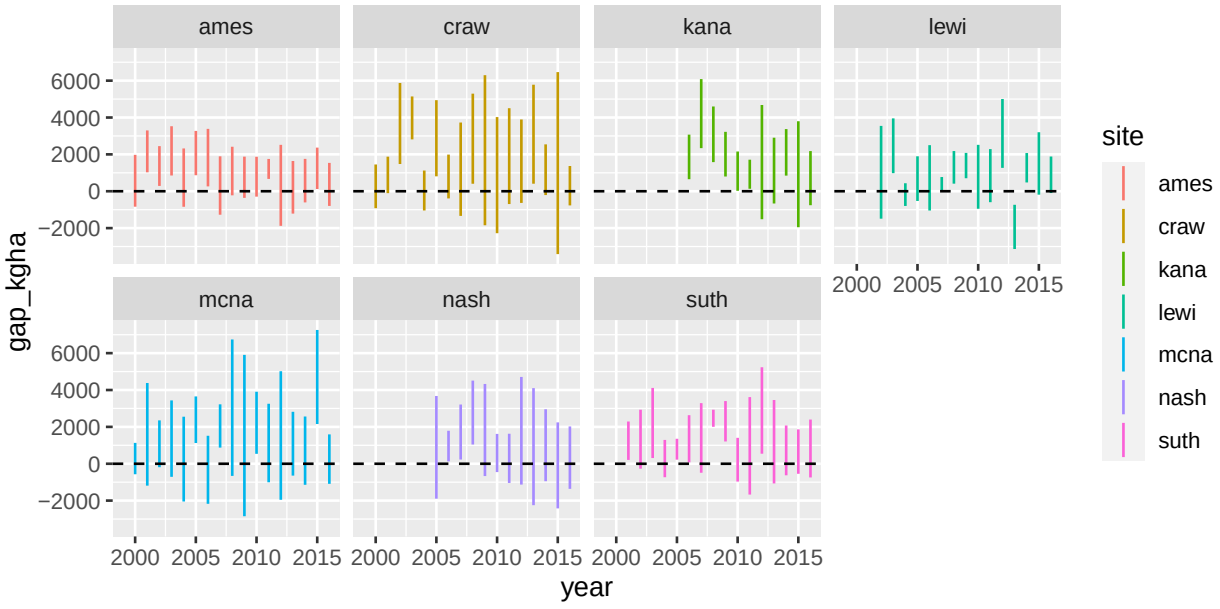
I think it might be easiest to use a Bonferroni-style adjustment to control the family-wise error rate (FWER): use α/n where n is the number of tests being done.

```
nrow(na.omit(dat2)) # 105 rows with complete data
```

```
## [1] 105
```

Thus, using a smaller significance level makes the family-wise probability of Type I error 0.05, or 1/20, across all 105 tests by widening intervals, making them slightly more likely to contain zero, and thus fewer will be significantly non-zero (but those are the ones that are most likely to be a false positive if the null hypothesis is true).

```
crit_val <- qnorm(p = 1 - (.05 / 105)/2)

ggplot(dat3, aes(x = year, y = gap_kgha)) + # reuse same plot code from above
  geom_linerange(aes(ymin = gap_kgha - crit_val*gap_se, # or use gap_se_site
                     ymax = gap_kgha + crit_val*gap_se, # or use gap_se_site
                     color = site)) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  facet_wrap(~site, nrow = 2) # changed to facet_wrap, personal preference :)
```



3/4. Power analysis

Following this resource, given any three of the following items, the fourth can be exactly determined:

1. sample size
2. effect size
3. significance level = P(Type I error) = probability of finding an effect that is not there *(usually .05)*
4. power = 1 - P(Type II error) = probability of finding an effect that is there *(commonly .8 or .95)*

The `pwr` package will let us conduct a power analysis by providing three of the four items above and calculating the fourth. However, it takes effect size as Cohen's d, which is the difference of the two group means, contained in `dat2$gap_kgha`, divided by the pooled standard deviation, contained in `dat2$sp`.

```
pwr::pwr.t.test(n = 4, sig.level = .05, power = .8)
```

```
##
##      Two-sample t test power calculation
##
##               n = 4
##               d = 2.380757
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

According to these results, when two groups both contain 4 observations, we have an 80% chance of correctly detecting a true difference of 2.38 or larger and a 5% chance of falsely claiming a true difference when none exists.

In context, say a pair (or site) has $s_p = 500$, then we can calculate the mean difference associated with $d = 2.38$:

$$d = 2.38 = \frac{1190}{500}$$

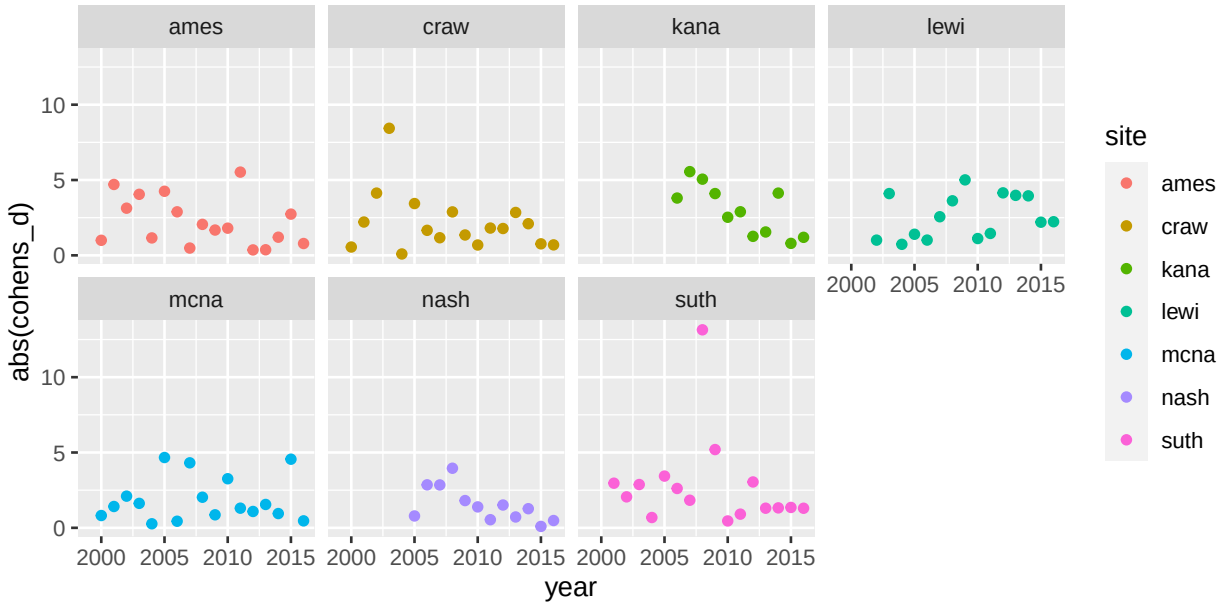
We can compute the same under the more strict conditions of power = .95 and $\alpha = .01$

```
pwr::pwr.t.test(n = 4, sig.level = .01, power = .95)
```

```
##
##      Two-sample t test power calculation
##
##               n = 4
##               d = 4.233368
##       sig.level = 0.01
##           power = 0.95
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

You can also go in reverse, and use a hypothetical effect size to determine the required sample size. First, let's look at the effect sizes in the data:

```
dat_d <- dat3 %>% mutate(cohens_d = gap_kgha / sp)

ggplot(dat_d) +
  geom_point(aes(x = year, y = abs(cohens_d), color = site)) +
  facet_wrap(~site, nrow = 2)
```

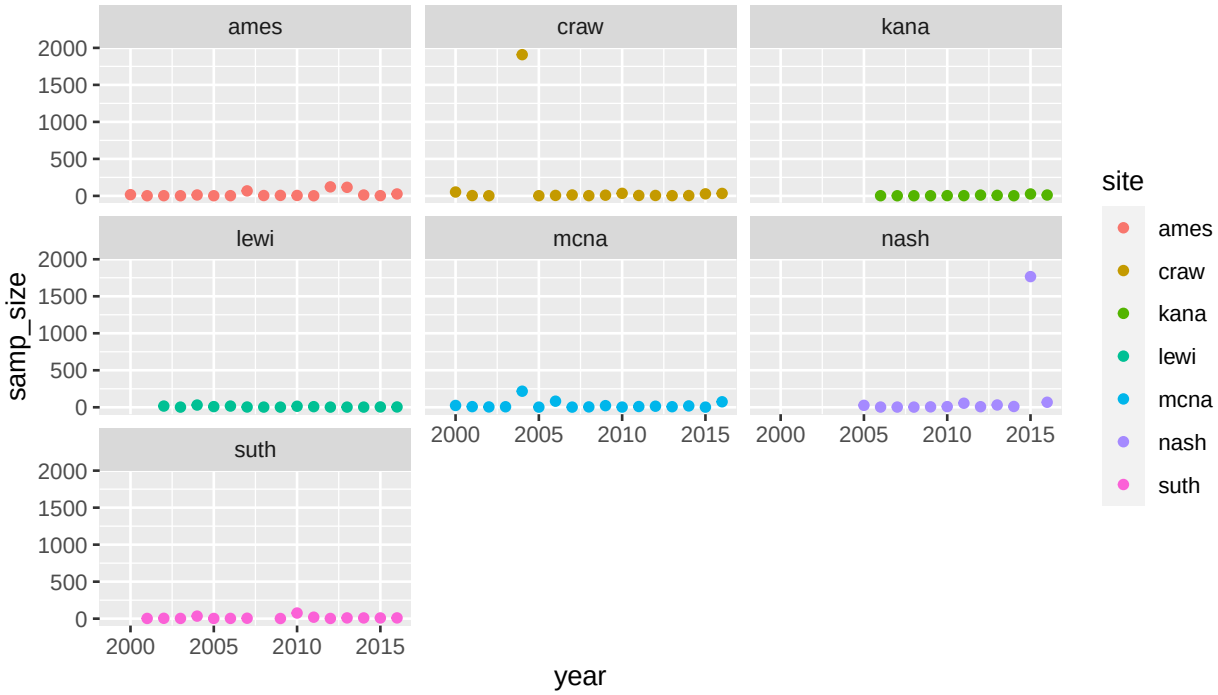


I'll use the common values of $\alpha = .05$, power $= .8$:

```
dat_d$samp_size <- sapply(abs(dat_d$cohens_d), function(d) {

  if(!is.na(d)) {

    out <- tryCatch({ # weird error with some d values
      pwr::pwr.t.test(sig.level = .05, power = .8, type = "two", d = d)
    }, error = function(e) { return(NA) })

    # At this point, `out` is either a list or NA from weird error
    # If list, return out$n
    if(any(is.na(out))) { return (NA)} else {return(out$n)}

  } else { return(NA) } # if d was NA to start
})
```

After figuring this out, I found `stats::power.t.test()` which lets you provide the means and sds without converting them to Cohen's d. You're welcome to adapt this code to use that function instead!

```
ggplot(dat_d) + geom_point(aes(x = year, y = samp_size, color = site)) +
  facet_wrap(~site)
```




```
# again with smaller y-lim
ggplot(dat_d) + geom_point(aes(x = year, y = samp_size, color = site)) +
  facet_wrap(~site) +
  coord_cartesian(ylim = c(0, 50))
```