ANCOVA

Neil J. Hatfield

4/18/2022

In this tutorial, we are going to explore Analysis of Covariance (ANCOVA) in R. The general structure for this guide will be:

- Setting Up R
  - Loading Packages, Setting Options, and Additional Tools
  - A New Package, `rstatix`
- Data Context
  - Repetitive Motion Disorders (Keyboarding)
- Fit the Model
  - Appropriateness of ANCOVA
  - Hasse Diagrams
  - Form the Model
- Assessing Assumptions
  - Gaussian Residuals
  - Homoscedasticity
  - Independence of Observations
  - Linear Relationship between Response and Covariate
  - Checking for Potential Outliers
  - Homogeneity of Slopes
- Results
  - Omnibus
  - Point Estimates
  - Post Hoc–Pairwise and Effect Sizes
- Choosing Amongst the Five ANCOVA Models

Setting Up R

Just as in the prior guides/tutorials, we have to first ensure that `R` is properly configured and prepared for our work. We will want to ensure that we load all of the appropriate packages, set our constraint, and load in any additional tools.

The core set of packages to load in include the following:

- `tidyverse`–for data cleaning/wrangling and the pipe, `%>%`
- `knitr` & `kableExtra`–for making professional looking tables

- `parameters`–to construct modern ANOVA tables, get omnibus effect sizes, and to switch out the type of *Sums of Squares*
- `hasseDiagram`–to construct Hasse diagrams
- `car` for nice QQ Plots
- `psych`–for easy descriptive statistics by group
- `emmeans`–for getting point estimates which attend to our models as well as doing post hoc analyses
- `rstatix`–our new package for detecting potential multivariate outliers

We are going to make use of one new package, `rstatix` for ANCOVA models. There is a function in this package, `mahalanobis_distance` that will help us with checking for any potential outliers.

As a reminder, the following code does all of these things:

```
# Demo code to set up R
## Load packages
packages <- c("tidyverse", "knitr", "kableExtra",
              "parameters", "hasseDiagram", "car",
              "psych", "emmeans", "rstatix")
lapply(packages, library, character.only = TRUE)

## Set options and constraint
options(knitr.kable.NA = "")
options(contrasts = c("contr.sum", "contr.poly"))

## Load useful tools
source("https://raw.github.com/neilhatfield/STAT461/master/rScripts/ANOVATools.R")
```

Data Context

When people engage in repetitive motion, they can suffer from any one of a set of Repetitive Motion Disorders such as carpal tunnel syndrome, trigger finger, or tendinitis. We are wanting to understand the impact of the type of keyboard on how many hours of pain a person experiences in their hands, wrists, and forearms. We suspect that the number of hours a person spends keyboarding is related to the number of hours of pain that they feel.

We have 12 volunteers who will use a specific keyboard we assign them for 2 weeks. During that time, they will record the number of hours they use the keyboard and the number of hours of repetitive motion pain during the study period.

Data

For this example, you'll want to import the data as shown below.

```
# Demo Code for loading data
# Keyboarding Situation
keyboardingData <- read.table(
  file = "https://raw.github.com/neilhatfield/STAT461/master/dataFiles/keyboarding.dat",
  header = TRUE,
  sep = ""
)

# Column Notes
```

```r
## hrs.pain is our response; hours experiencing pain
## kbd.type is our factor; type/style of keyboard
## hrs.kbd is our covariate; hours spent using the keyboard
### make sure that R is NOT thinking of hrs.kbd as factor but
### either as num or int

keyboardingData$kbd.type <- as.factor(keyboardingData$kbd.type)
```

Fit the Model

Before we write code to fit the ANCOVA model in R, we should check that ANCOVA is even appropriate.

Appropriateness of ANCOVA

To assess whether an ANCOVA model is even appropriate mimics that of any of our ANOVA models. That is,

1) Do we have a quantitative (ideally, continuous) response?
2) Do we have at least one qualitative/categorical factor of interest?
3) Do we have enough *Degrees of Freedom* to estimate all effects of interest?
4) Do we have enough *Degrees of Freedom* to estimate residuals/errors?
5) Are we aiming for an additive model (up to interaction terms)?

For ANCOVA, we add two more:

6) Do we have a quantitative attribute (i.e., a covariate) that we believe is related to our response?
7) Is there a linear relationship between our response and our covariate?

Just as you might suspect, we check all of these base requirements just as we have been: run through the study design, looking and verifying the elements. The Hasse diagram is our friend.

Hasse Diagrams

Speaking of Hasse diagrams, there is not an agreed upon convention for how to incorporate covariates into the diagram. I tend to think of the covariate as being similar to a block and thus place a new node at the second highest level of the diagram; next to our main effects and block. Since there are many (in some cases, infinitely) levels to a covariate, we don't put a number of levels out front (to the left). Rather, we'll put the label "cov" for covariate. The number of *Degrees of Freedom* will depend upon which which type of ANCOVA model you fit. However, if you are fitting the standard ANCOVA model ("parallel lines"/"separate intercepts"), then the covariate will use 1 *Degree of Freedom*.

Figure 1 shows the Hasse diagram for the Keyboarding study. Notice that our covariate is labelled and uses one *Degree of Freedom*. We still have positive values for the rest of the nodes' *Degrees of Freedom* thus we should be able to estimate the effects and errors.

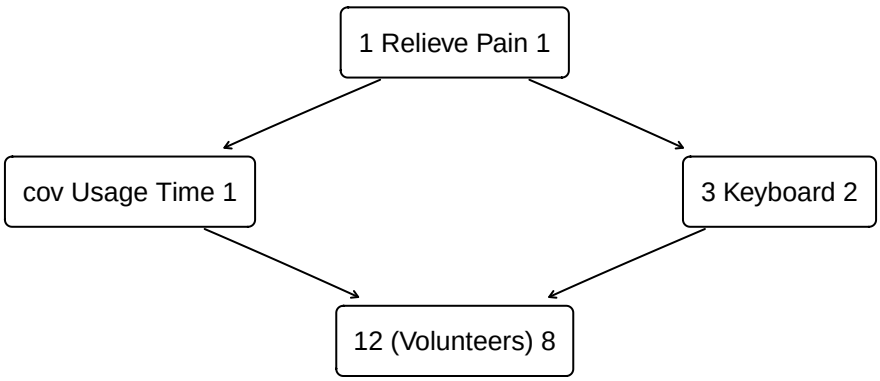


Figure 1: Hasse Diagram for Keyboarding Study

Linear Relationship

The last requirement for the ANCOVA is that there is a linear relationship between the response on the covariate. This requirement is also one of our assumptions. Thus, we'll look at this issue in that section.

Forming the Model

We are going to fit *two* models for ANCOVA: one will be our model that we use for checking all but two assumptions, and getting the results (both the omnibus and post hoc analyses). The other model we will use for checking the homogeneity of the covariate's slope parameter.

```
# Demo Code for Fitting ANCOVA
## Keyboarding Study
## Our core model
keyboardModel <- aov(
  formula = hrs.pain ~ hrs.kbd + kbd.type,
  data = keyboardingData
)

## Model for checking covariate's homogeneity
interactionCheck <- aov(
  formula = hrs.pain ~ hrs.kbd * kbd.type,
  data = keyboardingData
)
```

Exploring the Data

I want to leave a quick note as a reminder that just as we've discussed all course, you should explore your data as thoroughly as possible at this junction. Look at multiple data visualizations and at various values of descriptive statistics. This will not only help you build your understanding of the data (think, EDA) but will also set the stage for assessing several of our assumptions.

Assessing Assumptions

For ANCOVA models, we have six assumptions we need to check before using the parametric shortcut: our core three (Gaussian Residuals, Homoscedasticity, and Independent Observations) plus Linear Relationship between Response and Covariate, No Potential Outliers, and Homogeneity of Slopes.

We will start with the ANCOVA specific assumptions before returning to the core three.

Linear Relationship between Response and Covariate

The first assumption that I like to check in ANCOVA is that of the linear relationship between the covariate and the response. Again, the best approach here is to look at a scatter plot.

```
# Scatter plot of hours of pain and hours spent keyboarding
# Notice we're just using the data, not the model
ggplot(
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd
  )
) +
  geom_point(size = 2) +
  geom_smooth( # Adds a linear regression line
    inherit.aes = FALSE,
    mapping = aes(x = hrs.kbd, y = hrs.pain),
    method = "lm",
    formula = y ~ x,
    color = "black",
    linetype = "dashed",
    se = FALSE
  ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain")
```

Figure 2 shows the scatter plot of hours of pain experienced by the hours spent keyboarding. Notice that I've NOT incorporated the type of keyboard each person got assigned. I want to get as clear of a sense of the relationship between the response and the covariate as I possibly can. If I add too many other attributes to this plot (e.g., the factor), I might overlook what I need to attend to.

Ultimately, we're looking to see whether there is a linear relationship between the response and the covariate. That is to say, is the a constant rate of change between them. If we are plotting in the standard Cartesian coordinate system without scale transformations (e.g., log), then we should be able to sketch a straight line. In Figure 2, I added the plot of the estimated linear regression model of the response (hours of pain) with respect to the covariate (hours spent keyboarding). This graph appears as the dashed, black line. Does this line appear to mimic what we see in the scatter plot?

If we were to see a nonlinear relationship (e.g., a quadratic relationship), then I would go back to the model step and change the formula to `y ~ hrs.kbd^2 + kbd.type` (don't forget to update the `interactionCheck` too) to linearize the response-covariate relationship.

From Figure 2 I would say that we have a linear relationship between the response and our covariate.

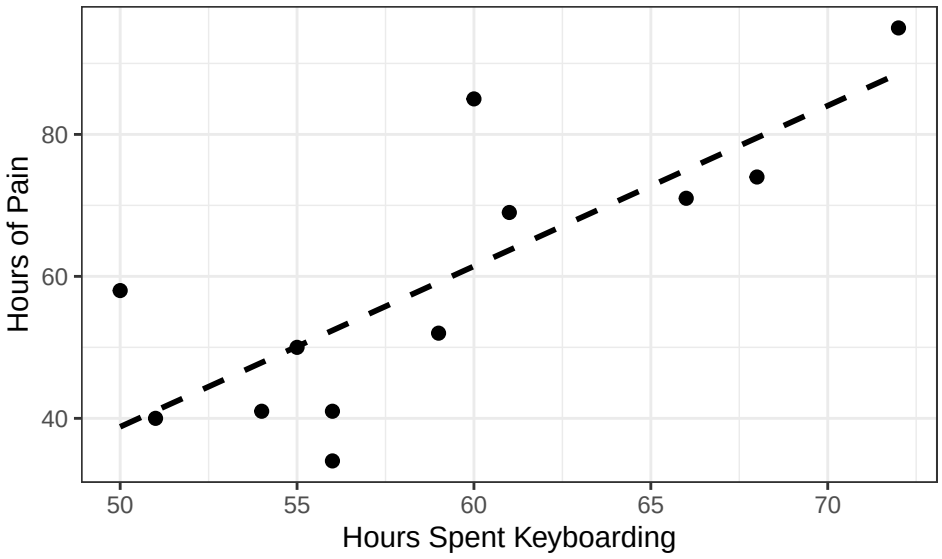


Figure 2: Hours of Pain vs Hours Spent Keyboarding

Checking Potential Outliers

While we always want to explore our data for potential outliers, ANCOVA models are less resistant to their influence than a typical balanced ANOVA model.

To explore for potential outliers, you will want to work in a systematic way, moving through your quantitative attributes one at a time. I recommend using Box plots and the various univariate Rules of Thumb to check.

```
# Demo Code for box plot with how to adjust the outlier detection
## Keyboarding Study
ggplot(
  data = keyboardingData,
  mapping = aes(x = hrs.pain)
) +
  geom_boxplot(
    coef = 1.5 # Use this to adjust outlier detection; coef*IQR
  ) +
  theme_void() +
  xlab("Hours of Pain") +
  theme(
    axis.line.x = element_line(),
    axis.text.x = element_text(size = 12),
    axis.title.x = element_text(size = 12)
  )
```

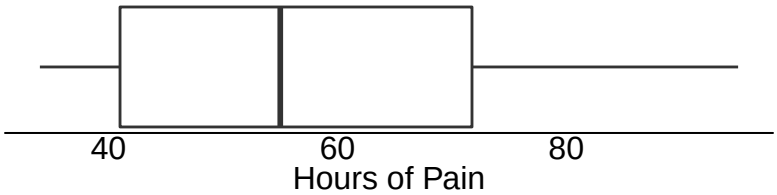


Figure 3: Box Plot of Hours of Pain

Once you complete all of the univariate checks for potential outliers, we can move on to *potential multivariate outliers*. A potential multivariate outlier is a case whose values appear to be inconsistent with the underlying structure of the rest of the collection. This is different from potential univariate outliers in that univariate are case's whose values tend to be extremes. A potential multivariate outlier may not be an outlier along any univariate check.

To help us check for potential multivariate outliers, we are going to make use of the `rstatix` package's `mahalanobis_distance` function.

```
# Demo Code for Detecting Multivariate Outliers
## Step 1: send the data through the Mahalanobis function
outlierDetection <- rstatix::mahalanobis_distance(keyboardingData)

## Step 2: OPTIONAL--reattach the factor
outlierDetection <- cbind(
  outlierDetection,
  factor = keyboardingData$kbd.type
)

## Step 3: Make a scatter plot
ggplot(
  data = outlierDetection,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    shape = is.outlier,
    color = factor
  )
) +
  geom_point(size = 3) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    color = "Keyboard",
    shape = "Potential Outlier"
  )
```

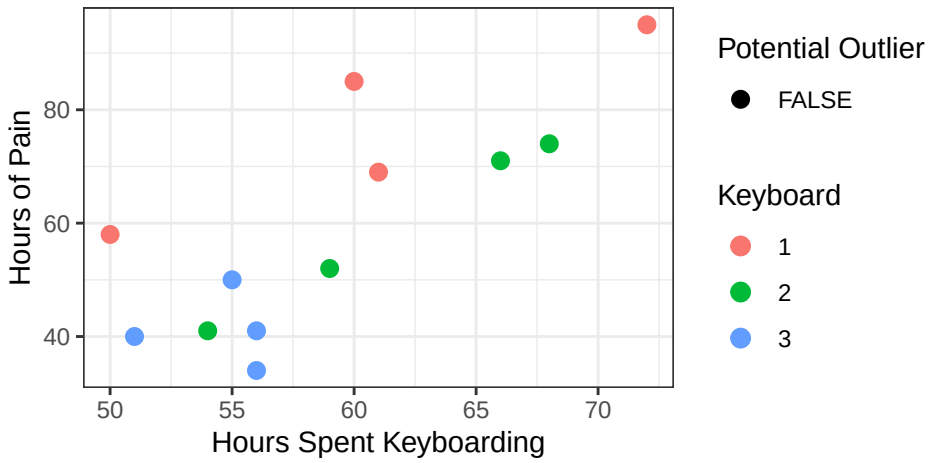


Figure 4: Potential Multivariate Outliers

In Figure 4, the shape of the points will reflect a `FALSE` or `TRUE` answer to the statement "This observation is a potential outlier." Thus, `TRUE` would indicate that we have a potential outlier. In our case, we have all `FALSE` points, thus we do not have a potential multivariate outliers to be concerned about.

If we had potential multivariate outliers, we would want to investigate why they are potential outliers. For example, is there a data entry error or some other mistake for those cases? If we can fix those errors, we should and then re-run our analyses. If we can't fix those errors, then we will want to consider removing those cases. If we do not know, then we should run the model with and without these cases and compare the results.

Homogeneity of Slopes

Standard ANCOVA models assume that the model is for separate intercepts/parallel lines for the covariate. We often refer to this as homogeneity of slopes. More accurately, this is the assumption that the rate of change of the response with respect to the covariate is invariant (unchanging) regarding the factor(s) (and block). Given that we're fitting a linear model between the response and the covariate, this means that we should have identical constant rates of change for each level of our factor(s).

There are two ways that we can assess this assumption: looking at how well a plot for the separate intercepts model fits our data and to do an informal test of the interaction between the covariate and our factor(s).

Plot

One route we can take is to build a plot of the response (amount of time in pain) by the covariate (time spent keyboarding) and the type of keyboard used. This is similar to what we did to check for the linear relationship between the response and the covariate. However, there are two distinct differences. First, we're going to explicitly include the factor (type of keyboard) to our plot. Second, we're going to impose three regression lines–one for each kind of keyboard–that only differ in their intercepts. If these lines appear to match up with the data reasonably well, we can be conveniced that we have homogeneity of slopes. If they do not, then we might need to re-think this assumption.

```
# Demo Code for Assessing Homogeneity of Slopes
ggplot(
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    color = kbd.type,
    shape = kbd.type
  )
) +
  geom_point(size = 2) +
  geom_smooth(
    method = "lm", # See notes below
    mapping = aes(y = predict(keyboardModel)),
    formula = y ~ x,
    se = FALSE
  ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    color = "Keyboard Type",
```

```
    shape = "Keyboard Type"
  )
```

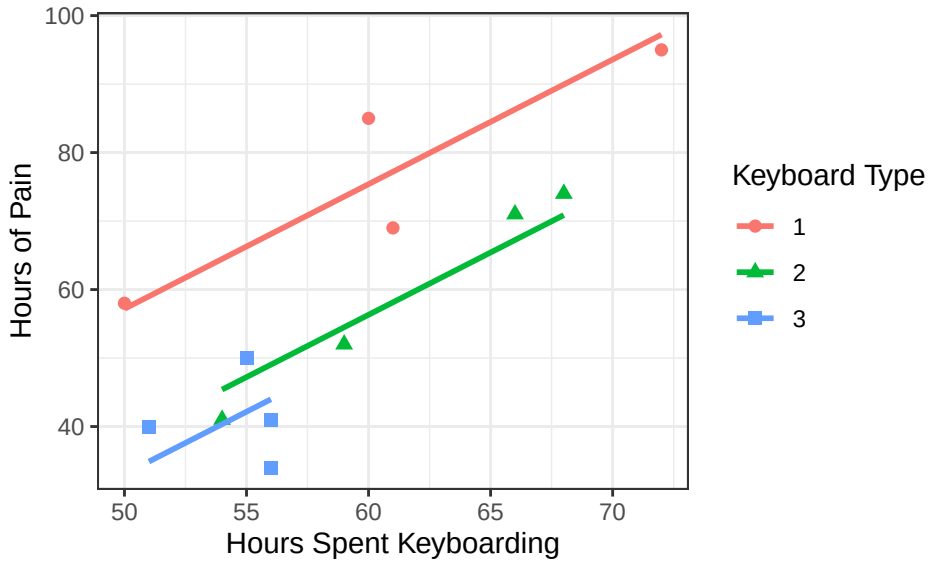


Figure 5: Homogeneity of Slopes

In the code for the plot, you'll notice that we used the `geom_smooth` geometry to add the three lines. For these lines we want to use a linear model (`method = "lm"`) and define the formula `y ~ x`. The most important aspect was that within `geom_smooth` we replaced the observed pain duration with the predicted duration from our ANCOVA model: `predict(keyboardModel)`.

I recommend checking out the section at the end of this guide to learn more about the five different ANCOVA models.

Informal Test of Interaction

The second approach we can take is to do an informal test of the interaction term. Recall that we formed a second ANCOVA model called `interactionCheck`. We will want to check this set of results using Type III *Sum of Squares*.

To get the information, we'll use the `car` package's `Anova` function to get the Type III *Sum of Squares*:

```
# Remember to use the car package
car::Anova(
  mod = interactionCheck,
  type = 3
)
```

```
## Anova Table (Type III tests)
##
## Response: hrs.pain
##                Sum Sq Df F value  Pr(>F)
## (Intercept)     18.807  1  0.3881 0.55622
## hrs.kbd        217.487  1  4.4880 0.07845 .
## kbd.type       147.651  2  1.5235 0.29171
```

```
## hrs.kbd:kbd.type 120.271  2  1.2410 0.35398
## Residuals        290.756  6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To do this informal check, you will want to keep your Unusualness Threshold and Type I Error Risk in mind. While we are going to look at an ANOVA table, we do not report this table. At most, you'll extract the information you need an report that in a paragraph, much like what we did for the Kruskal-Wallis H test.

We only want to focus on the interaction term of hours spent keyboarding and the type of keyboard (i.e., `hrs.kbd:kbd.type`). The p-value for this term is approximately 0.35. This is well beyond any UT you're allowed to use in this course, thus, we will fail to reject the null hypothesis. That is to say, the interaction between the covariate and the factor is not statistically important. This tells us that we should not anticipate different rates of change of pain duration with respect to time spent keyboarding for different kinds of keyboards.

Quick Word of Wisdom

You'll notice that even though I listed the core three assumptions first, I didn't start with checking them. This is due to the fact that if the linear relationship, potential outliers, and homogeneity of slopes are violated we need to address those issues before we check the core three. Addressing these three will directly impact the core assumptions. Thus, by checking these first, I can save time in the long run.

Gaussian Residuals

Use a QQ plot like usual:

```
# QQ plot for residuals
car::qqPlot(
  x = residuals(keyboardModel),
  distribution = "norm",
  envelope = 0.90,
  id = FALSE,
  pch = 20,
  ylab = "Residuals (hours)"
)
```

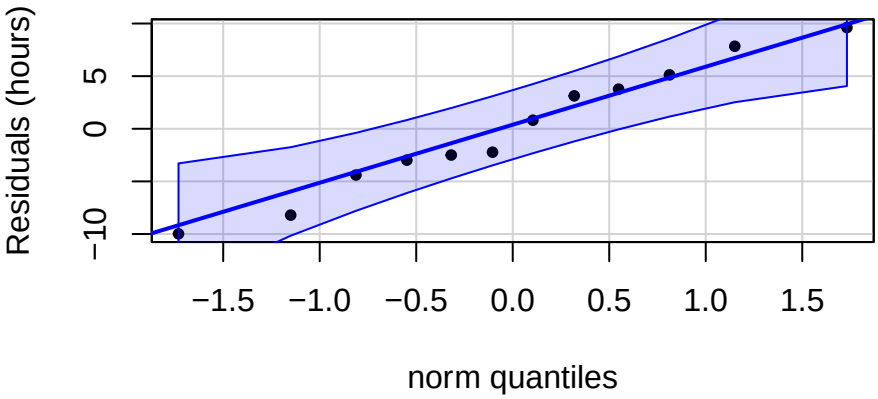


Figure 6: QQ Plot for Residuals

There is very little to be concerned about in our QQ plot (Figure 6); we will go ahead and proceed as if our residuals follow a Gaussian distribution.

Homoscedasticity

Given that we have more than just a single factor, we need to look at a Tukey-Anscombe plot rather than just a strip chart.

```
ggplot(
  data = data.frame(
    residuals = residuals(keyboardModel),
    fitted = fitted.values(keyboardModel)
  ),
  mapping = aes(x = fitted, y = residuals)
) +
  geom_point(size = 2) +
  geom_hline(
    yintercept = 0,
    linetype = "dashed",
    color = "grey50"
  ) +
  geom_smooth(
    formula = y ~ x,
    method = stats::loess,
    method.args = list(degree = 1),
    se = FALSE,
    size = 0.5
  ) +
  theme_bw() +
  xlab("Fitted values (hours)") +
  ylab("Residuals (hours)")
```

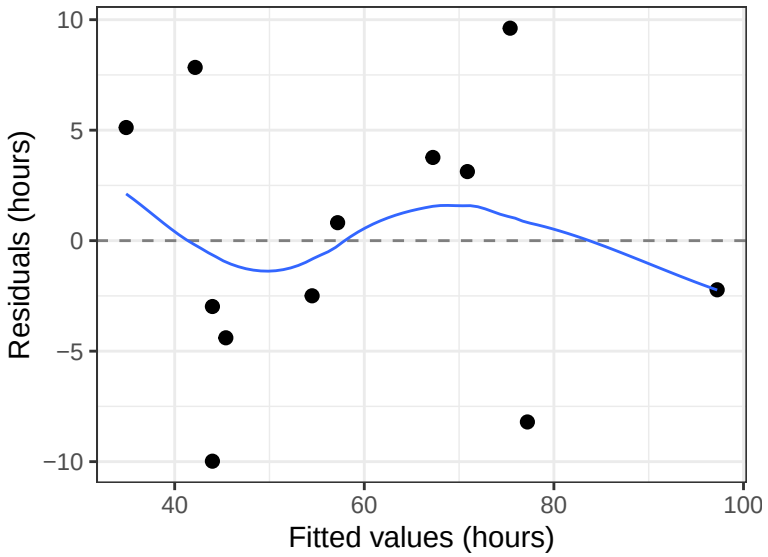


Figure 7: Tukey-Anscombe Plot for Keyboarding Study

I'm a bit hesitant for the homoscedasticity assumption given the curvy nature of the line (see Figure 7). However, I don't get the sense of any pattern to the residuals. So I might say that homoscedasticity might be questionable. I would anticipate that I would go back and look at box plots and descriptive statistics by my factor to see if there might be any numeric evidence of heteroscedasticity.

Independence of Observations

Unfortunately, we don't know measurement order so index plots are not going to be useful here. However, we can think through the study design and reach the decision that we have independent observations.

(I'm leaving this to each of you to practice and come up with a justification for why we can say that we have independence of observations.)

Additional Checks for ANCOVA

We do have more assumptions to check in an ANCOVA situation than an ANOVA situation. Keep in mind that if you introduce a Block or Random effects, then those assumptions will get imported into your situation as well. That is, block doesn't interact with factors, and each random effect factor has treatment effects which follow a Gaussian distribution. The approaches you used previously (as well as much of the code) remains the same.

Multiple Covariates

If you are planning on using multiple covariates in your model, then there is one extra assumption that you need to check. We assume that that all of our covariates are not highly correlated with one another; that is, we do not want to have multicollinearity amongst our covariates.

There are two tools you can use to check out the possibility of multicolinearity. The first is the Generalized Variance Inflation Factor and the second is the Squared Multiple Correlation.

For the Generalized Variance Inflation Factor (GVIF), you can use the `car::vif` function on the ANCOVA model. While, you'll get GVIF values for all terms in the model, we can focus on the covariates. Look at the column labelled `GVIF^(1/(2*Df))` and square the values there. This will give a value that you can compare to the typical VIF Rules of Thumb for 5 and 10.

For the Squared Multiple Correlation (SMC), we can make use of the function `psych::smc`. For this function, you'll use all of the covariate columns from the data frame as the input. You'll get back a listing of SMC values for each covariate. The Rule of Thumb here is that values beyond 0.5 indicate that we might have redundancy (multicollinearity).

Results

Results for ANCOVA models are a bit mixed: some people will only report Omnibus Results, others will proceed to looking at Post Hoc analyses. This is to say, there's not an overriding drive to automatically conduct Post Hoc analysis like there is for a CRD/One-way layout. My suggestion is to let the research questions posed guide you.

Omnibus Results

In this particular situation, we have a **balanced** design, thus we do not need to worry about different types of Sums of Squares.

```
# Omnibus Test/Modern ANOVA Table
parameters::model_parameters(
    model = keyboardModel,
    omega_squared = "partial",
    eta_squared = "partial",
    epsilon_squared = "partial"
) %>%
  dplyr::mutate( #Fixing the Parameter (Source) Column's values
    Parameter = dplyr::case_when(
      Parameter == "hrs.kbd" ~ "Hours Spent Keyboarding",
      Parameter == "kbd.type" ~ "Keyboard Type",
      TRUE ~ Parameter
    )
  ) %>%
  knitr::kable(
    digits = 4,
  col.names = c("Source", "SS", "df", "MS", "F", "p-value",
                "Partial Omega Sq.", "Partial Eta Sq.", "Partial Epsilon Sq."),
  caption = "ANOVA Table for Keyboarding Study",
  align = c('l',rep('c',8)),
  booktab = TRUE
) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 12,
    latex_options = c("scale_down", "HOLD_position")
  )
```

Table 1: ANOVA Table for Keyboarding Study

Source	SS	df	MS	F	p-value	Partial Omega Sq.	Partial Eta Sq.	Partial Epsilon Sq.
Hours Spent Keyboarding	2598.8209	1	2598.8209	50.5819	0.0001	0.8051	0.8634	0.8464
Keyboard Type	1195.8180	2	597.9090	11.6373	0.0043	0.6394	0.7442	0.6803
Residuals	411.0278	8	51.3785					

While we don't necessarily want to ignore the covariate's row in the table, we don't want to get caught up in that row. That is, don't forget that our focus is on our factor(s). Our interpretations remain the same as before.

- Point Estimates
- Post Hoc–Pairwise and Effect Sizes
- Choosing Amongst the Five ANCOVA Models

Post Hoc Analysis

Point Estimates I want to quickly remind you that you can get point estimates for your main effects and treatment effects using the `dummy.coef` function. If you need confidence intervals for these, you can use the `confint` function (don't forget to provide an *adjusted* confidence level).

```
# Point Estimates for Keyboarding Model
## Don't use raw output in your reports, make a nice table
dummy.coef(keyboardModel)
```

```
## Full coefficients are
##
## (Intercept):     -48.2072
## hrs.kbd:          1.819896
## kbd.type:                1         2         3
##                 14.398515 -4.671381 -9.727135
```

Notice that our intercept is negative! For the ANCOVA models, the `(Intercept)` is NO LONGER the *Grand Mean*. To get the estimate of the *Grand Mean* you will need to take the intercept value and add 1.819896 * the *SAM* of the covariate (which is 59): $-48.2072 + 1.819896 * 59 = 59.17$ hours per person.

To get the the covariate-adjusted point estimates (i.e., the marginal cell means), we will turn to the `emmeans` package.

```
## Use emmeans to get the appropriate margins AND do Pairwise
emmOut <- emmeans::emmeans(
  object = keyboardModel,
  specs = pairwise ~ kbd.type,
  adjust = "tukey",
  level = 0.9
)

## Point Estimates
as.data.frame(emmOut$emmeans) %>%
  knitr::kable(
    digits = 4,
    col.names = c("Keyboard Type", "Marginal Mean","SE", "DF",
                  "Lower Bound","Upper Bound"),
    caption = "Marginal Means-Tukey 90\\% Adjustment",
    align = c("l", rep("c", 5)),
    booktabs = TRUE
  ) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 12,
    latex_options = c("HOLD_position")
  )
```

Table 2: Marginal Means-Tukey 90% Adjustment

Keyboard Type	Marginal Mean	SE	DF	Lower Bound	Upper Bound
1	73.5652	3.6406	8	66.7953	80.3350
2	54.4953	3.7223	8	47.5736	61.4170
3	49.4395	3.9434	8	42.1066	56.7725

Pairwise Comparisons Just as with Factorial models, we will use the `emmeans` package here as well. Since I've already stored the output, I don't need to call `emmeans` a second time. I had forgotten to mention that the using `$contrasts` on the output of `emmeans` will give you when I made the Factorial Designs document.

```
# Pairwise Comparisons
as.data.frame(emmOut$contrasts) %>%
  knitr::kable(
    digits = 4,
    col.names = c("Comparison", "Difference","SE", "DF",
                  "t Statistic","p-value"),
    caption = "Marginal Means-Tukey 90\\% Adjustment",
    align = c("l", rep("c", 5)),
    booktabs = TRUE
  ) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 12,
    latex_options = c("HOLD_position")
  )
```

Table 3: Marginal Means-Tukey 90% Adjustment

Comparison	Difference	SE	DF	t Statistic	p-value
1 - 2	19.0699	5.0816	8	3.7527	0.0138
1 - 3	24.1257	5.5596	8	4.3395	0.0062
2 - 3	5.0558	5.7195	8	0.8839	0.6647

The `adjust` argument of `emmeans` allows for the following values for confidence intervals: `"bonferroni"`, `"tukey"`, `"scheffe"`, and `"sidak"`. If you do not want confidence intervals you may use values of `"holm"`, `"hochberg"`, `"hommel"`, `"BH"` (Benjamini and Hochberg), and `"fdr"`.

Effect Sizes

You will want to use the `emmeans` package for the effect sizes as well (the `anova.PostHoc` function won't account for the covariate).

```
# Pass the stored marginals into the effect size function
cohenType <- emmeans::eff_size(
  object = emmOut$emmeans,
  sigma = sigma(keyboardModel),
  edf = df.residual(keyboardModel)
)

# Create a data frame, add on the probability of superiority
# Send that data frame into a nice table
as.data.frame(cohenType) %>%
  dplyr::mutate(
    ps = probSup(effect.size),
    .after = effect.size
  ) %>%
  dplyr::select(contrast, effect.size, ps) %>%
  knitr::kable(
    digits = 3,
    col.names = c("Comparison", "Cohen's d", "Probability of Superiority"),
```

```
    align = "lcc",
    caption = "Effect Sizes for Keyboard Type",
    booktab = TRUE
  ) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 12,
    latex_options = "HOLD_position"
  )
```

Table 4: Effect Sizes for Keyboard Type

Comparison	Cohen's d	Probability of Superiority
1 - 2	2.660	0.970
1 - 3	3.366	0.991
2 - 3	0.705	0.691

Five ANCOVA Models

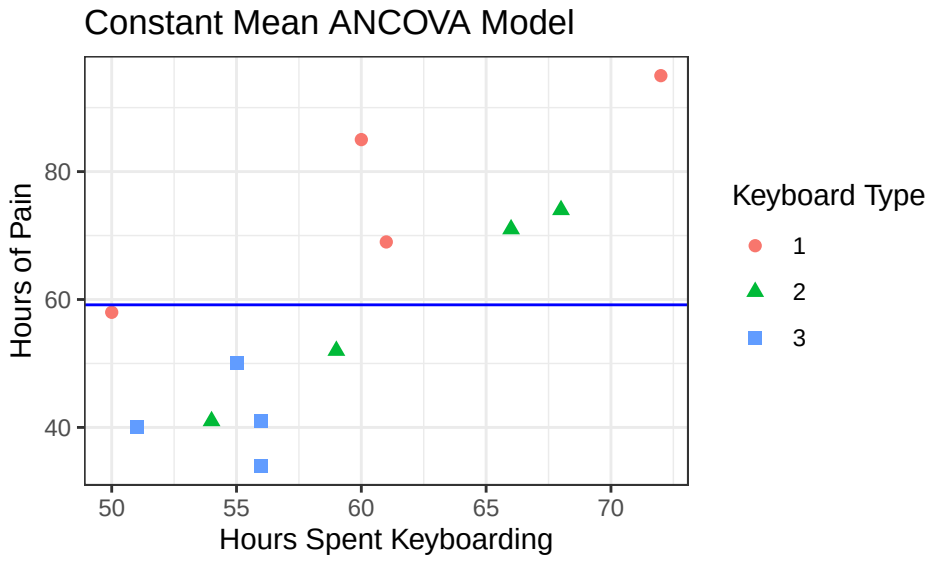
There are actually five (5) different ANCOVA models. Each one is a slight variation and is applicable in different situations. They all have impacts on the *degrees of freedom* within the model, so you'll want to be cautious.

No Effects–Constant [Grand] Mean

This is the null model: that our factor(s) and our covariate(s) have absolutely no impact on the response. If this were true, we would end up with the following plot:

```
# Constant Mean
ggplot(
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    color = kbd.type,
    shape = kbd.type
  )
) +
  geom_point(size = 2) +
  geom_hline(
    yintercept = mean(keyboardingData$hrs.pain),
    color = "blue"
  ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    title = "Constant Mean ANCOVA Model",
    color = "Keyboard Type",
```

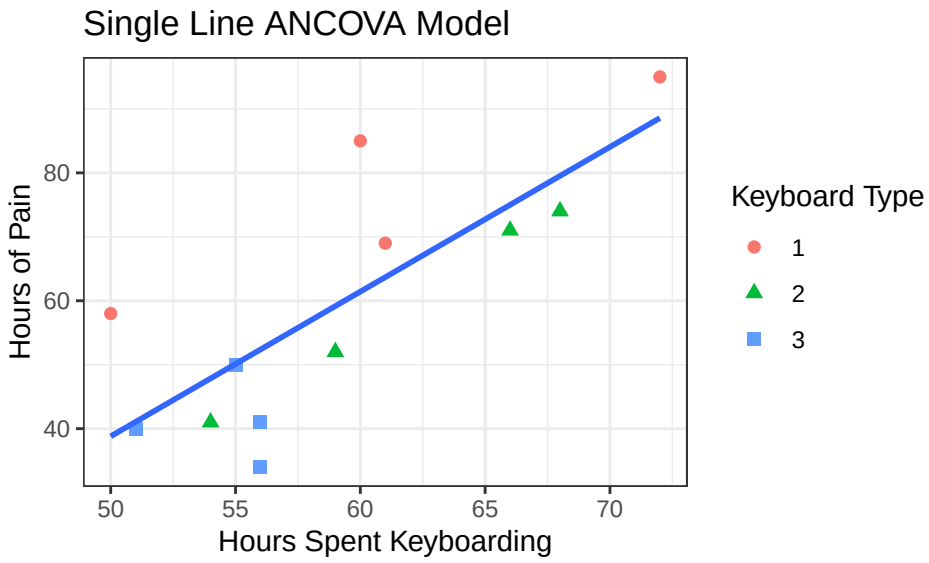
```
    shape = "Keyboard Type"
  )
```



Single Line

This ANCOVA model assumes that the covariate(s) affect the response but the factor(s) does not. Thus, there is a single line for the relationship.

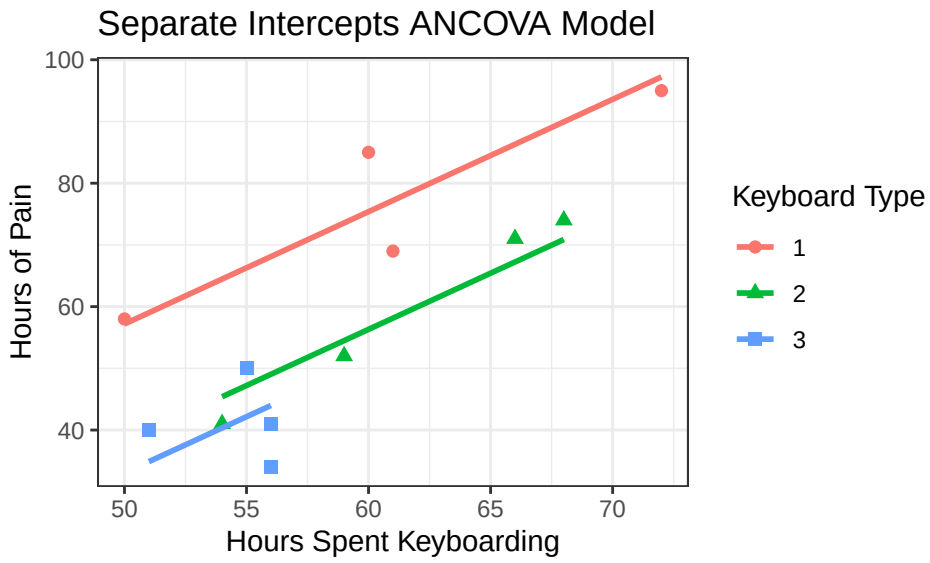
```
# Single Line
ggplot(
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    color = kbd.type,
    shape = kbd.type
  )
) +
  geom_point(size = 2) +
  geom_smooth(
    inherit.aes = FALSE,
    mapping = aes(x = hrs.kbd, y = hrs.pain),
    method = "lm",
    formula = y ~ x,
    se = FALSE
    ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    title = "Single Line ANCOVA Model",
    color = "Keyboard Type",
    shape = "Keyboard Type"
  )
```



Separate Intercept/Parallel Lines

This is the standard ANCOVA model and the one that we most often want to draw upon. (This is the model that we are using in this course.)

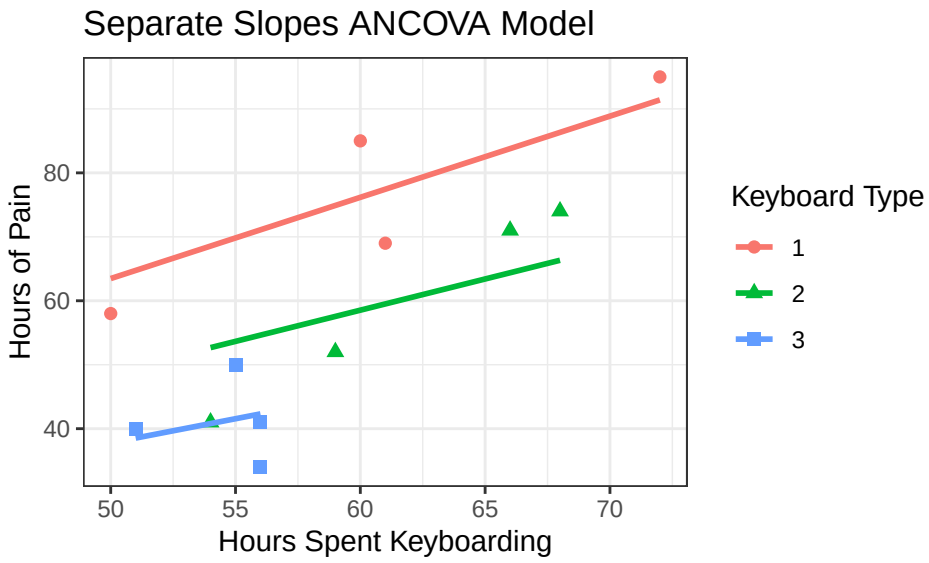
```
# Separate Intercepts
ggplot(
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    color = kbd.type,
    shape = kbd.type
  )
) +
  geom_point(size = 2) +
  geom_smooth(
    method = "lm",
    mapping = aes(y = predict(keyboardModel)),
    formula = y ~ x,
    se = FALSE
  ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    title = "Separate Intercepts ANCOVA Model",
    color = "Keyboard Type",
    shape = "Keyboard Type"
  )
```



Separate Slopes

This version of the ANCOVA model fixes the y-intercept to the same value for all groups but allows each group to have their own constant rate of change (slopes) for the covariate term.

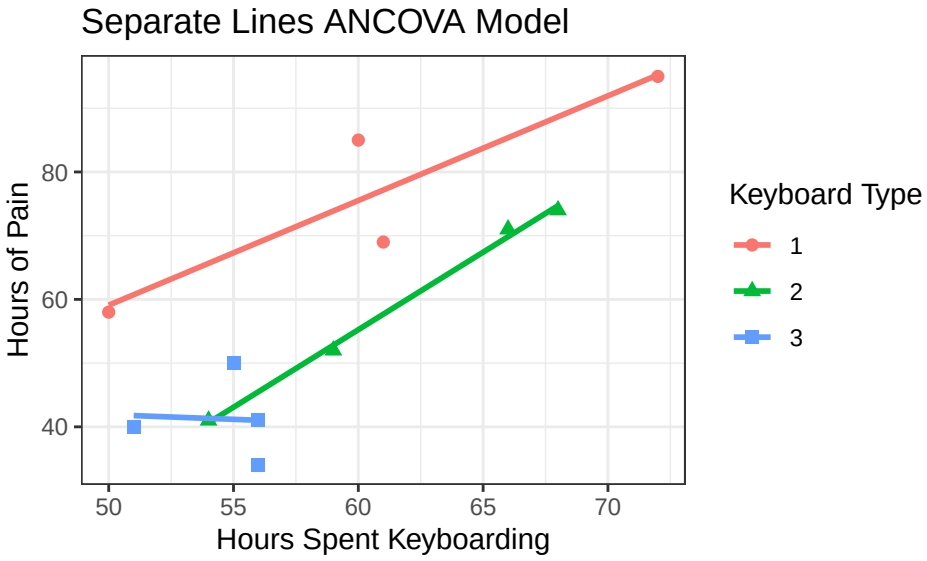
```
# Separate Slopes
ggplot(
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    color = kbd.type,
    shape = kbd.type
  )
) +
  geom_point(size = 2) +
  geom_smooth(
    method = "lm",
    formula = y ~ x + 0,
    se = FALSE
  ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    title = "Separate Slopes ANCOVA Model",
    color = "Keyboard Type",
    shape = "Keyboard Type"
  )
```



Separate Lines

The last ANCOVA model is that of separate lines; here is where we believe that there is a theoretically important reason to have the factor(s) and covariate(s) interact. We could see this with a statistically significant interaction term. For our keyboarding data, we would see the following.

```
# Separate Lines
ggplot(
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    color = kbd.type,
    shape = kbd.type
  )
) +
  geom_point(size = 2) +
  geom_smooth(
    method = "lm",
    formula = y ~ x,
    se = FALSE
  ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    title = "Separate Lines ANCOVA Model",
    color = "Keyboard Type",
    shape = "Keyboard Type"
  )
```



Notice that each of these models looks different within our Keyboarding context. We can also see that some models appear to be inconsistent with our data (e.g., the Constant Mean ANCOVA model). You can learn more about these models in Section 17.3 of Oehlert.

Code Appendix

```
# Setting Document Options
knitr::opts_chunk$set(
  echo = FALSE,
  warning = FALSE,
  message = FALSE,
  fig.align = "center"
)

packages <- c("tidyverse", "knitr", "kableExtra",
              "parameters", "hasseDiagram", "car",
              "psych", "emmeans", "rstatix")
lapply(packages, library, character.only = TRUE)

options(knitr.kable.NA = "")
options(contrasts = c("contr.sum", "contr.poly"))

source("https://raw.github.com/neilhatfield/STAT461/master/rScripts/ANOVATools.R")

# Demo code to set up R
## Load packages
packages <- c("tidyverse", "knitr", "kableExtra",
              "parameters", "hasseDiagram", "car",
              "psych", "emmeans", "rstatix")
lapply(packages, library, character.only = TRUE)

## Set options and constraint
options(knitr.kable.NA = "")
options(contrasts = c("contr.sum", "contr.poly"))

## Load useful tools
source("https://raw.github.com/neilhatfield/STAT461/master/rScripts/ANOVATools.R")

# Demo Code for loading data
# Keyboarding Situation
keyboardingData <- read.table(
  file = "https://raw.github.com/neilhatfield/STAT461/master/dataFiles/keyboarding.dat",
  header = TRUE,
  sep = ""
)

# Column Notes
## hrs.pain is our response; hours experiencing pain
## kbd.type is our factor; type/style of keyboard
## hrs.kbd is our covariate; hours spent using the keyboard
### make sure that R is NOT thinking of hrs.kbd as factor but
### either as num or int

keyboardingData$kbd.type <- as.factor(keyboardingData$kbd.type)

# Demo Code for Hasse diagram
## Keyboarding Study
```

```
modelLabels <- c("1 Relieve Pain 1", "cov Usage Time 1",
                 "3 Keyboard 2", "12 (Volunteers) 8")
modelMatrix <- matrix(
  data = c(FALSE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE,
           FALSE, TRUE, FALSE, FALSE, FALSE, TRUE, TRUE,
           TRUE, FALSE),
  nrow = 4,
  ncol = 4,
  byrow = FALSE
)
hasseDiagram::hasse(
 data = modelMatrix,
 labels = modelLabels
)

# Demo Code for Fitting ANCOVA
## Keyboarding Study
## Our core model
keyboardModel <- aov(
  formula = hrs.pain ~ hrs.kbd + kbd.type,
  data = keyboardingData
)

## Model for checking covariate's homogeneity
interactionCheck <- aov(
  formula = hrs.pain ~ hrs.kbd * kbd.type,
  data = keyboardingData
)

# Scatter plot of hours of pain and hours spent keyboarding
# Notice we're just using the data, not the model
ggplot(
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd
  )
) +
  geom_point(size = 2) +
  geom_smooth( # Adds a linear regression line
    inherit.aes = FALSE,
    mapping = aes(x = hrs.kbd, y = hrs.pain),
    method = "lm",
    formula = y ~ x,
    color = "black",
    linetype = "dashed",
    se = FALSE
  ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain")

# Demo Code for box plot with how to adjust the outlier detection
```

```r
## Keyboarding Study
ggplot(
  data = keyboardingData,
  mapping = aes(x = hrs.pain)
) +
  geom_boxplot(
    coef = 1.5 # Use this to adjust outlier detection; coef*IQR
  ) +
  theme_void() +
  xlab("Hours of Pain") +
  theme(
    axis.line.x = element_line(),
    axis.text.x = element_text(size = 12),
    axis.title.x = element_text(size = 12)
  )

# Demo Code for Detecting Multivariate Outliers
## Step 1: send the data through the Mahalanobis function
outlierDetection <- rstatix::mahalanobis_distance(keyboardingData)

## Step 2: OPTIONAL--reattach the factor
outlierDetection <- cbind(
  outlierDetection,
  factor = keyboardingData$kbd.type
)

## Step 3: Make a scatter plot
ggplot(
  data = outlierDetection,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    shape = is.outlier,
    color = factor
  )
) +
  geom_point(size = 3) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    color = "Keyboard",
    shape = "Potential Outlier"
  )


# Demo Code for Assessing Homogeneity of Slopes
ggplot(
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    color = kbd.type,
```

```
    shape = kbd.type
  )
) +
  geom_point(size = 2) +
  geom_smooth(
    method = "lm", # See notes below
    mapping = aes(y = predict(keyboardModel)),
    formula = y ~ x,
    se = FALSE
  ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    color = "Keyboard Type",
    shape = "Keyboard Type"
  )

# Remember to use the car package
car::Anova(
  mod = interactionCheck,
  type = 3
)

# QQ plot for residuals
car::qqPlot(
  x = residuals(keyboardModel),
  distribution = "norm",
  envelope = 0.90,
  id = FALSE,
  pch = 20,
  ylab = "Residuals (hours)"
)


ggplot(
  data = data.frame(
    residuals = residuals(keyboardModel),
    fitted = fitted.values(keyboardModel)
  ),
  mapping = aes(x = fitted, y = residuals)
) +
  geom_point(size = 2) +
  geom_hline(
    yintercept = 0,
    linetype = "dashed",
    color = "grey50"
  ) +
  geom_smooth(
    formula = y ~ x,
    method = stats::loess,
    method.args = list(degree = 1),
    se = FALSE,
```

```
    size = 0.5
  ) +
  theme_bw() +
  xlab("Fitted values (hours)") +
  ylab("Residuals (hours)")

# Omnibus Test/Modern ANOVA Table
parameters::model_parameters(
    model = keyboardModel,
    omega_squared = "partial",
    eta_squared = "partial",
    epsilon_squared = "partial"
) %>%
  dplyr::mutate( #Fixing the Parameter (Source) Column's values
    Parameter = dplyr::case_when(
      Parameter == "hrs.kbd" ~ "Hours Spent Keyboarding",
      Parameter == "kbd.type" ~ "Keyboard Type",
      TRUE ~ Parameter
    )
  ) %>%
  knitr::kable(
    digits = 4,
  col.names = c("Source", "SS", "df", "MS", "F", "p-value",
                "Partial Omega Sq.", "Partial Eta Sq.", "Partial Epsilon Sq."),
  caption = "ANOVA Table for Keyboarding Study",
  align = c('l',rep('c',8)),
  booktab = TRUE
) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 12,
    latex_options = c("scale_down", "HOLD_position")
  )

# Point Estimates for Keyboarding Model
## Don't use raw output in your reports, make a nice table
dummy.coef(keyboardModel)

## Use emmeans to get the appropriate margins AND do Pairwise
emmOut <- emmeans::emmeans(
  object = keyboardModel,
  specs = pairwise ~ kbd.type,
  adjust = "tukey",
  level = 0.9
)

## Point Estimates
as.data.frame(emmOut$emmeans) %>%
  knitr::kable(
    digits = 4,
    col.names = c("Keyboard Type", "Marginal Mean","SE", "DF",
                  "Lower Bound","Upper Bound"),
    caption = "Marginal Means-Tukey 90\\% Adjustment",
```

```
    align = c("l", rep("c", 5)),
    booktabs = TRUE
  ) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 12,
    latex_options = c("HOLD_position")
  )

# Pairwise Comparisons
as.data.frame(emmOut$contrasts) %>%
  knitr::kable(
    digits = 4,
    col.names = c("Comparison", "Difference","SE", "DF",
                  "t Statistic","p-value"),
    caption = "Marginal Means-Tukey 90\\% Adjustment",
    align = c("l", rep("c", 5)),
    booktabs = TRUE
  ) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 12,
    latex_options = c("HOLD_position")
  )

# Pass the stored marginals into the effect size function
cohenType <- emmeans::eff_size(
  object = emmOut$emmeans,
  sigma = sigma(keyboardModel),
  edf = df.residual(keyboardModel)
)

# Create a data frame, add on the probability of superiority
# Send that data frame into a nice table
as.data.frame(cohenType) %>%
  dplyr::mutate(
    ps = probSup(effect.size),
    .after = effect.size
  ) %>%
  dplyr::select(contrast, effect.size, ps) %>%
  knitr::kable(
    digits = 3,
    col.names = c("Comparison", "Cohen's d", "Probability of Superiority"),
    align = "lcc",
    caption = "Effect Sizes for Keyboard Type",
    booktab = TRUE
  ) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 12,
    latex_options = "HOLD_position"
  )
```

```r
# Constant Mean
ggplot(
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    color = kbd.type,
    shape = kbd.type
  )
) +
  geom_point(size = 2) +
  geom_hline(
    yintercept = mean(keyboardingData$hrs.pain),
    color = "blue"
  ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    title = "Constant Mean ANCOVA Model",
    color = "Keyboard Type",
    shape = "Keyboard Type"
  )

# Single Line
ggplot(
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    color = kbd.type,
    shape = kbd.type
  )
) +
  geom_point(size = 2) +
  geom_smooth(
    inherit.aes = FALSE,
    mapping = aes(x = hrs.kbd, y = hrs.pain),
    method = "lm",
    formula = y ~ x,
    se = FALSE
    ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    title = "Single Line ANCOVA Model",
    color = "Keyboard Type",
    shape = "Keyboard Type"
  )

# Separate Intercepts
ggplot(
```

```
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    color = kbd.type,
    shape = kbd.type
  )
) +
  geom_point(size = 2) +
  geom_smooth(
    method = "lm",
    mapping = aes(y = predict(keyboardModel)),
    formula = y ~ x,
    se = FALSE
  ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    title = "Separate Intercepts ANCOVA Model",
    color = "Keyboard Type",
    shape = "Keyboard Type"
  )

# Separate Slopes
ggplot(
  data = keyboardingData,
  mapping = aes(
    y = hrs.pain,
    x = hrs.kbd,
    color = kbd.type,
    shape = kbd.type
  )
) +
  geom_point(size = 2) +
  geom_smooth(
    method = "lm",
    formula = y ~ x + 0,
    se = FALSE
  ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    title = "Separate Slopes ANCOVA Model",
    color = "Keyboard Type",
    shape = "Keyboard Type"
  )

# Separate Lines
ggplot(
  data = keyboardingData,
  mapping = aes(
```

```
    y = hrs.pain,
    x = hrs.kbd,
    color = kbd.type,
    shape = kbd.type
  )
) +
  geom_point(size = 2) +
  geom_smooth(
    method = "lm",
    formula = y ~ x,
    se = FALSE
  ) +
  theme_bw() +
  xlab("Hours Spent Keyboarding") +
  ylab("Hours of Pain") +
  labs(
    title = "Separate Lines ANCOVA Model",
    color = "Keyboard Type",
    shape = "Keyboard Type"
  )
```