CURE_SkillBuilding

Abby Beatty

March 15, 2020

Read in packages and SE function

Summary Function

```
summarySE <- function(data=NULL, measurevar, groupvars=NULL, na.rm=TRUE,
                      conf.interval=.95, .drop=TRUE) {
    library(plyr)
    # New version of length which can handle NA's: if na.rm==T, don't count them
    length2 <- function (x, na.rm=TRUE) {
        if (na.rm) sum(!is.na(x))
        else       length(x)
    }
    # This does the summary. For each group's data frame, return a vector with
    # N, mean, and sd
    datac <- ddply(data, groupvars, .drop=.drop,
      .fun = function(xx, col) {
        c(N    = length2(xx[[col]], na.rm=na.rm),
          mean = mean   (xx[[col]], na.rm=na.rm),
          sd   = sd     (xx[[col]], na.rm=na.rm)
        )
      },
      measurevar
    )
    # Rename the "mean" column
    datac <- rename(datac, c("mean" = measurevar))
    datac$se <- datac$sd / sqrt(datac$N)  # Calculate standard error of the mean
    # Confidence interval multiplier for standard error
    # Calculate t-statistic for confidence interval:
    # e.g., if conf.interval is .95, use .975 (above/below), and use df=N-1
    ciMult <- qt(conf.interval/2 + .5, datac$N-1)
    datac$ci <- datac$se * ciMult
    return(datac)
    }
```

18.19 post data: Confidence Measures

```
model=read.csv("Models.csv")
model$Experience=factor(model$Experience, levels=c("Guided", "Autonomous"))
model$Time=factor(model$Time, levels=c("Pre", "Post"))
```

```
conf=subset(model, Skill == "Confidence")

conf.lm=lm(Score~ Experience*Subskill, data=conf, na.action=na.omit)
anova(conf.lm)
```

```
## Analysis of Variance Table
##
## Response: Score
##                      Df  Sum Sq Mean Sq F value    Pr(>F)
## Experience            1   1.217  1.2168  1.3302   0.24957
## Subskill              3  71.057 23.6856 25.8934 4.06e-15 ***
## Experience:Subskill   3   8.604  2.8681  3.1354   0.02564 *
## Residuals           342 312.839  0.9147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
conf.means=emmeans(conf.lm, list(pairwise ~ Experience|Subskill), adjust = "tukey")
conf.means
```

```
## $`emmeans of Experience | Subskill`
## Subskill = Design:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       3.25 0.133 342     2.99     3.51
##  Autonomous   3.12 0.137 342     2.85     3.39
##
## Subskill = Notebook:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.13 0.133 342     3.87     4.40
##  Autonomous   4.29 0.137 342     4.02     4.55
##
## Subskill = Perform:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.21 0.133 342     3.95     4.47
##  Autonomous   4.18 0.137 342     3.91     4.45
##
## Subskill = Weakness:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.08 0.195 342     3.70     4.47
##  Autonomous   3.22 0.199 342     2.83     3.61
##
## Confidence level used: 0.95
##
## $`pairwise differences of Experience | Subskill`
## Subskill = Design:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous   0.1276 0.190 342   0.670  0.5034
##
## Subskill = Notebook:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous  -0.1511 0.190 342  -0.794  0.4280
##
## Subskill = Perform:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous   0.0279 0.190 342   0.146  0.8837
##
```

```
## Subskill = Weakness:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous   0.8659 0.279 342   3.103  0.0021
```

```
conf.plot=plot(conf.means, by = "Subskill", comparisons = TRUE, horizontal = FALSE, color="darkgreen", 
  theme_bw()
```

18.19 post data: Applicabilty Measures

```
app=subset(model, Skill == "Applicability")

app.lm=lm(Score~Experience*Subskill, data=app, na.action=na.omit)
anova(app.lm)
```

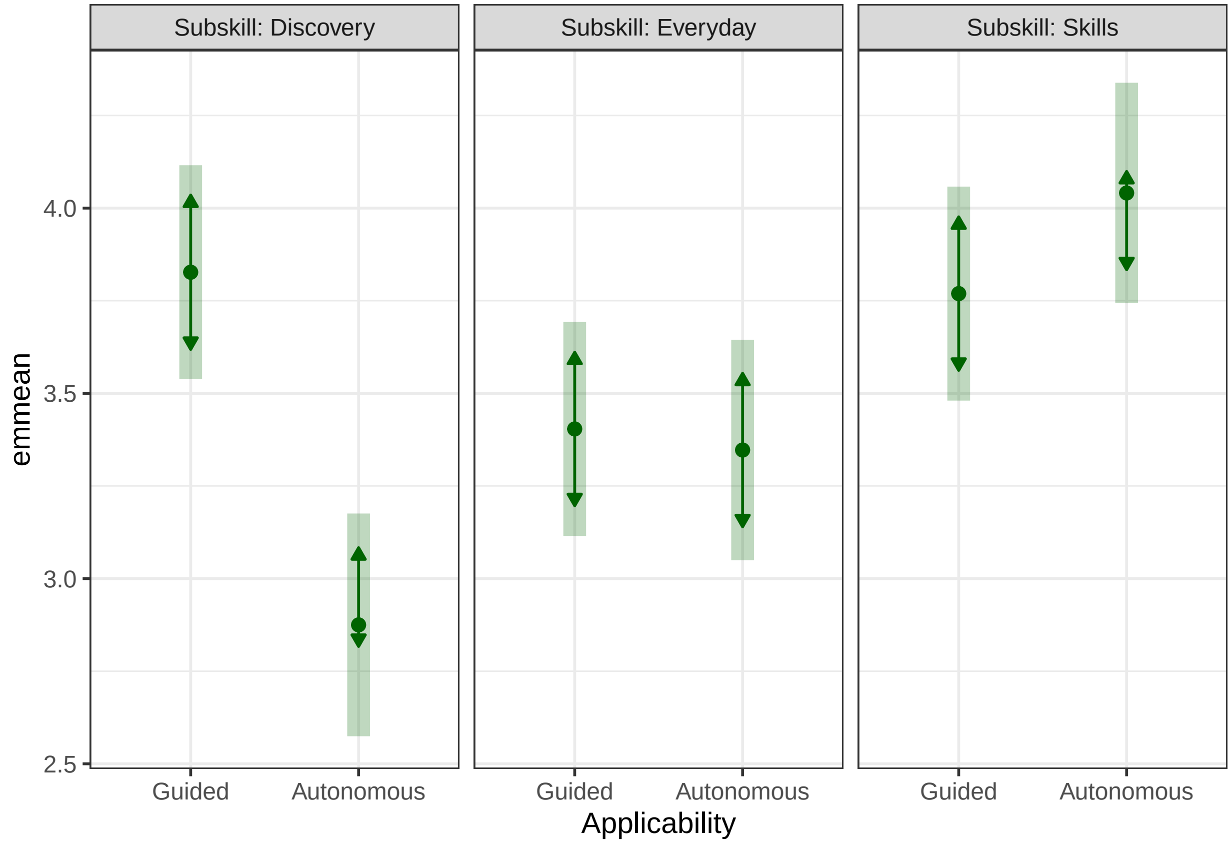
```
## Analysis of Variance Table
##
## Response: Score
##                      Df Sum Sq Mean Sq F value    Pr(>F)
## Experience            1   4.42  4.4171  3.9445 0.0479472 *
## Subskill              2  18.78  9.3885  8.3841 0.0002872 ***
## Experience:Subskill   2  20.10 10.0490  8.9739 0.0001645 ***
## Residuals           296 331.46  1.1198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
app.means=emmeans(app.lm, list(pairwise ~ Experience|Subskill), cov.keep="Year", adjust = "tukey")
app.means
```

```
## $`emmeans of Experience | Subskill`
## Subskill = Discovery:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       3.83 0.147 296     3.54     4.12
##  Autonomous   2.88 0.153 296     2.57     3.18
##
## Subskill = Everyday:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       3.40 0.147 296     3.12     3.69
##  Autonomous   3.35 0.151 296     3.05     3.64
##
## Subskill = Skills:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       3.77 0.147 296     3.48     4.06
##  Autonomous   4.04 0.151 296     3.74     4.34
##
## Confidence level used: 0.95
##
## $`pairwise differences of Experience | Subskill`
## Subskill = Discovery:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous   0.9519 0.212 296   4.494  <.0001
##
## Subskill = Everyday:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous   0.0569 0.211 296   0.270  0.7873
```

```
##
## Subskill = Skills:
##  contrast             estimate    SE  df t.ratio p.value
##  Guided - Autonomous  -0.2716 0.211 296 -1.289  0.1984
```

```
app.plot=plot(app.means, by = "Subskill", comparisons = TRUE, horizontal = FALSE, color="darkgreen", xla
  theme_bw()


app.plot
```



18.19 post data: CURE Measures

```
cure=subset(model, Skill == "CURE")

cure.lm=lm(Score~Experience*Subskill, data=cure, na.action=na.omit)
anova(cure.lm)
```

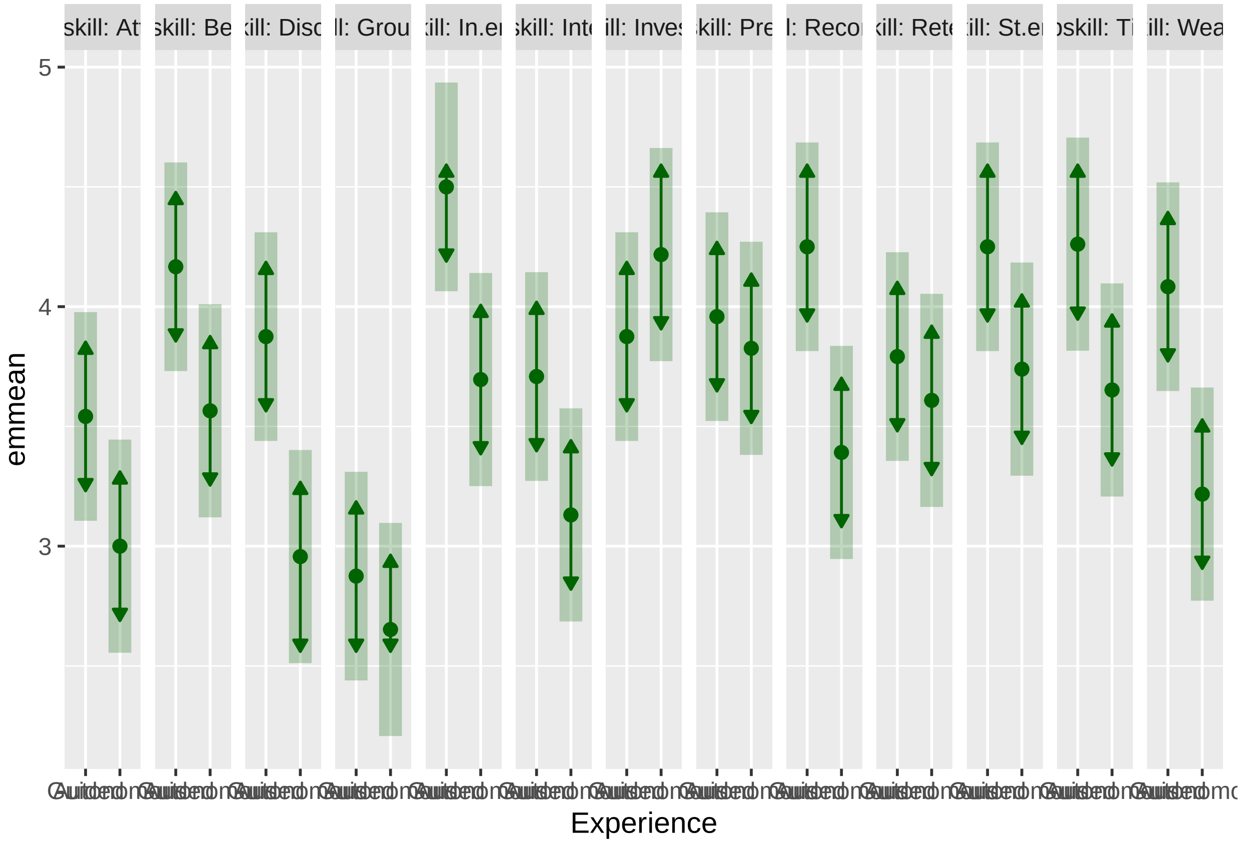
```
## Analysis of Variance Table
##
## Response: Score
##                     Df Sum Sq Mean Sq F value    Pr(>F)
## Experience           1  37.76  37.759 31.9887 2.430e-08 ***
## Subskill            12  81.12   6.760  5.7271 2.181e-09 ***
## Experience:Subskill 12  18.58   1.549  1.3119     0.207
## Residuals           584 689.35   1.180
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cure.means=emmeans(cure.lm, list(pairwise ~ Experience|Subskill), adjust = "tukey")
cure.means
```

```
## $`emmeans of Experience | Subskill`
## Subskill = Attend:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       3.54 0.222 584     3.11     3.98
##  Autonomous   3.00 0.227 584     2.56     3.44
##
## Subskill = Benefit:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.17 0.222 584     3.73     4.60
##  Autonomous   3.57 0.227 584     3.12     4.01
##
## Subskill = Discovery:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       3.88 0.222 584     3.44     4.31
##  Autonomous   2.96 0.227 584     2.51     3.40
##
## Subskill = Group Work:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       2.88 0.222 584     2.44     3.31
##  Autonomous   2.65 0.227 584     2.21     3.10
##
## Subskill = In.engage:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.50 0.222 584     4.06     4.94
##  Autonomous   3.70 0.227 584     3.25     4.14
##
## Subskill = Interest:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       3.71 0.222 584     3.27     4.14
##  Autonomous   3.13 0.227 584     2.69     3.58
##
## Subskill = Investment:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       3.88 0.222 584     3.44     4.31
##  Autonomous   4.22 0.227 584     3.77     4.66
##
## Subskill = Prepare:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       3.96 0.222 584     3.52     4.39
##  Autonomous   3.83 0.227 584     3.38     4.27
##
## Subskill = Recommend:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.25 0.222 584     3.81     4.69
##  Autonomous   3.39 0.227 584     2.95     3.84
##
## Subskill = Retention:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       3.79 0.222 584     3.36     4.23
##  Autonomous   3.61 0.227 584     3.16     4.05
##
## Subskill = St.engage:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.25 0.222 584     3.81     4.69
##  Autonomous   3.74 0.227 584     3.29     4.18
##
## Subskill = Time:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.26 0.227 584     3.82     4.71
##  Autonomous   3.65 0.227 584     3.21     4.10
##
## Subskill = Weakness:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.08 0.222 584     3.65     4.52
##  Autonomous   3.22 0.227 584     2.77     3.66
##
## Confidence level used: 0.95
##
## $`pairwise differences of Experience | Subskill`
## Subskill = Attend:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous    0.542 0.317 584  1.709  0.0881
##
## Subskill = Benefit:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous    0.601 0.317 584  1.897  0.0583
##
## Subskill = Discovery:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous    0.918 0.317 584  2.897  0.0039
##
## Subskill = Group Work:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous    0.223 0.317 584  0.703  0.4824
##
## Subskill = In.engage:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous    0.804 0.317 584  2.537  0.0114
##
## Subskill = Interest:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous    0.578 0.317 584  1.823  0.0688
##
## Subskill = Investment:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous   -0.342 0.317 584 -1.080  0.2806
##
## Subskill = Prepare:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous    0.132 0.317 584  0.417  0.6767
##
## Subskill = Recommend:
```

```
##   contrast              estimate    SE  df t.ratio p.value
##   Guided - Autonomous      0.859 0.317 584   2.709  0.0070
##
## Subskill = Retention:
##   contrast              estimate    SE  df t.ratio p.value
##   Guided - Autonomous      0.183 0.317 584   0.577  0.5641
##
## Subskill = St.engage:
##   contrast              estimate    SE  df t.ratio p.value
##   Guided - Autonomous      0.511 0.317 584   1.611  0.1076
##
## Subskill = Time:
##   contrast              estimate    SE  df t.ratio p.value
##   Guided - Autonomous      0.609 0.320 584   1.900  0.0579
##
## Subskill = Weakness:
##   contrast              estimate    SE  df t.ratio p.value
##   Guided - Autonomous      0.866 0.317 584   2.731  0.0065
```

```
plot(cure.means, by = "Subskill", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )
```



```
#subset only significant ones for plot
cure.sub=subset(cure, Subskill == "Time"| Subskill == "Benefit"| Subskill == "Recommend" | Subskill ==

curesub.lm=lm(Score~Experience*Subskill, data=cure.sub, na.action=na.omit)
anova(curesub.lm)
```

```
## Analysis of Variance Table
##
## Response: Score
##                      Df  Sum Sq Mean Sq F value    Pr(>F)
## Experience             1  33.594  33.594 29.7356 1.305e-07 ***
## Subskill               4  12.177   3.044  2.6946    0.0318 *
## Experience:Subskill   4   0.991   0.248  0.2193    0.9275
## Residuals           224 253.067   1.130
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cure.sub.means=emmeans(curesub.lm, list(pairwise ~ Experience|Subskill), adjust = "tukey")
cure.sub.means
```

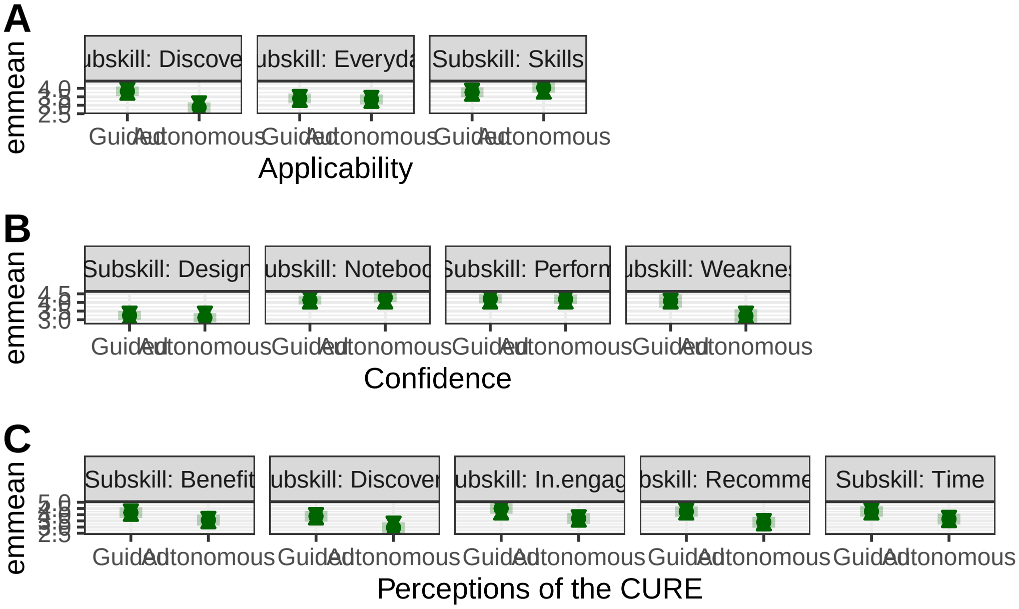
```
## $`emmeans of Experience | Subskill`
## Subskill = Benefit:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.17 0.217 224     3.74     4.59
##  Autonomous   3.57 0.222 224     3.13     4.00
##
## Subskill = Discovery:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       3.88 0.217 224     3.45     4.30
##  Autonomous   2.96 0.222 224     2.52     3.39
##
## Subskill = In.engage:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.50 0.217 224     4.07     4.93
##  Autonomous   3.70 0.222 224     3.26     4.13
##
## Subskill = Recommend:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.25 0.217 224     3.82     4.68
##  Autonomous   3.39 0.222 224     2.95     3.83
##
## Subskill = Time:
##  Experience emmean    SE  df lower.CL upper.CL
##  Guided       4.26 0.222 224     3.82     4.70
##  Autonomous   3.65 0.222 224     3.22     4.09
##
## Confidence level used: 0.95
##
## $`pairwise differences of Experience | Subskill`
## Subskill = Benefit:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous    0.601 0.310 224 1.939   0.0537
##
## Subskill = Discovery:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous    0.918 0.310 224 2.961   0.0034
##
## Subskill = In.engage:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous    0.804 0.310 224 2.593   0.0101
##
```

```
## Subskill = Recommend:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous    0.859 0.310 224 2.769   0.0061
##
## Subskill = Time:
##  contrast            estimate    SE  df t.ratio p.value
##  Guided - Autonomous    0.609 0.313 224 1.942   0.0534
```

```
cure.plot=plot(cure.sub.means, by = "Subskill", comparisons = TRUE, horizontal = FALSE, color="darkgreen
  theme_bw()
```

draw multi panel plot

```
p=ggdraw() +
  draw_plot(app.plot, x = 0, y = 0.66, width = 0.6, height = .28) +
  draw_plot(conf.plot, x = 0, y = .33, width = 0.8, height = .28)+
  draw_plot(cure.plot, x = 0, y = 0.0, width = 1, height = 0.28) +
   draw_plot_label(label = c("C", "B","A"), size = 15,
                  x = c(0, 0 ,0), y = c(0.33, 0.66,0.99))


p2= p + theme(plot.margin=unit(c(15,15,15,15),"mm"))

ggsave(p2, file="multipanel.png", width=10, height=12, dpi=600)

p2
```



18.19 pre vs post data: Confidence Measures

```
#Pre and post comparison in 2018
conf.18=subset(conf, Experience == "Guided")
conf.18=subset(conf.18, Subskill != "Weakness")

conf.18pp=lme(Score~as.factor(Time)*Subskill, random= ~1|ID, data=conf.18, na.action=na.omit)

anova(conf.18pp)
```

```
##                           numDF denDF  F-value p-value
## (Intercept)                   1   123 909.9595  <.0001
## as.factor(Time)               1   123   1.7778  0.1849
## Subskill                      2   123  30.6972  <.0001
## as.factor(Time):Subskill      2   123   0.7653  0.4674
```

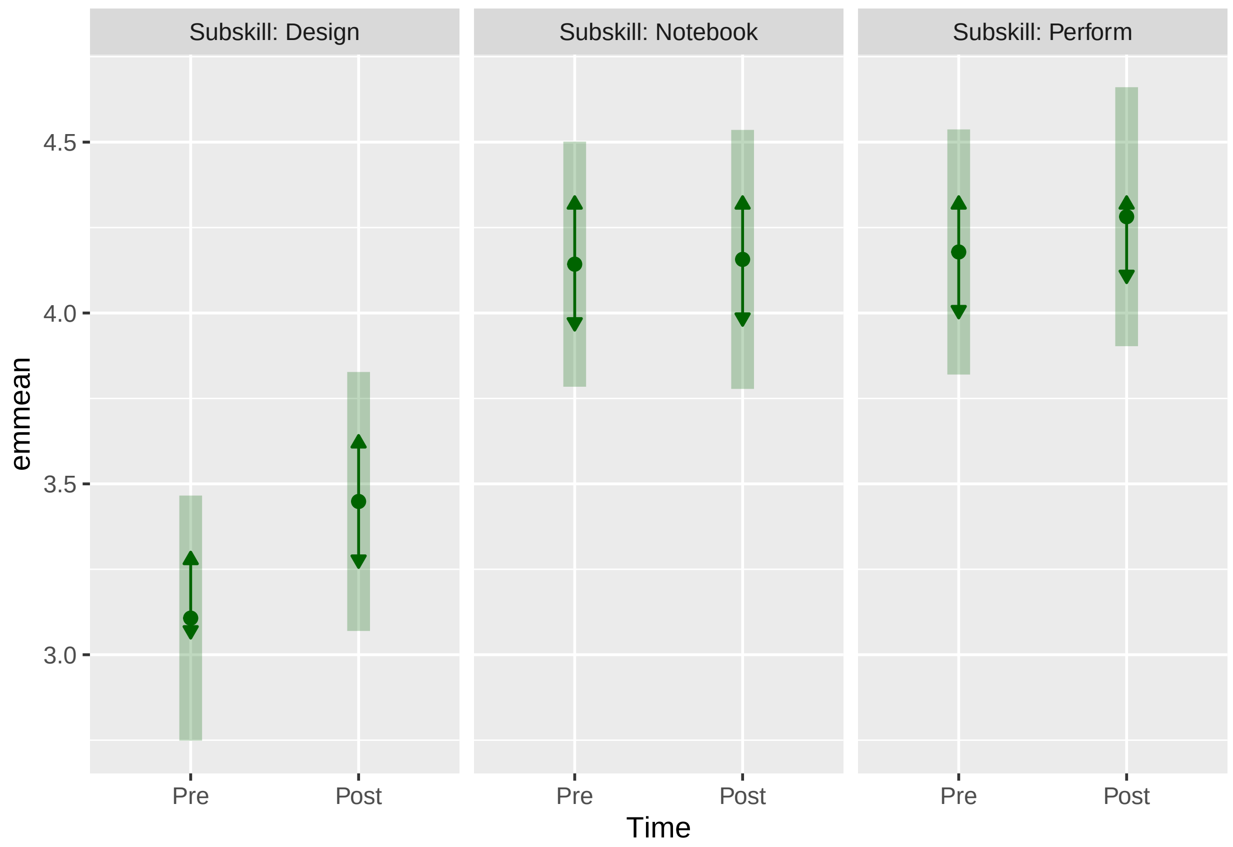
```
confmeans18=emmeans(conf.18pp, list(pairwise ~ Time|Subskill), adjust = "tukey")
```

```
## Warning in model.frame.default(formula, data = data, ...): variable 'Time' is
## not a factor
```

```
confmeans18
```

```
## $`emmeans of Time | Subskill`
## Subskill = Design:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    3.11 0.175 27     2.75     3.47
##  Post   3.45 0.185 27     3.07     3.83
##
## Subskill = Notebook:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    4.14 0.175 27     3.78     4.50
##  Post   4.16 0.185 27     3.78     4.54
##
## Subskill = Perform:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    4.18 0.175 27     3.82     4.54
##  Post   4.28 0.185 27     3.90     4.66
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $`pairwise differences of Time | Subskill`
## Subskill = Design:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post   -0.341 0.195 123 -1.749  0.0828
##
## Subskill = Notebook:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post   -0.014 0.195 123 -0.072  0.9430
##
## Subskill = Perform:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post   -0.103 0.195 123 -0.529  0.5977
##
## Degrees-of-freedom method: containment
```

```r
plot(confmeans18, by = "Subskill", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )
```



```
#Pre and post comparison in 2019
conf.19=subset(conf, Experience == "Autonomous")
conf.19=subset(conf.19, Subskill != "Weakness")


conf.19pp=lme(Score~as.factor(Time)*Subskill, random= ~1|ID, data=conf.19, na.action=na.omit)

anova(conf.19pp)
```

```
##                           numDF denDF   F-value p-value
## (Intercept)                   1   109 1681.3206  <.0001
## as.factor(Time)               1   109    2.3479  0.1284
## Subskill                      2   109   31.8675  <.0001
## as.factor(Time):Subskill      2   109    3.2425  0.0429
```

```
confmeans19=emmeans(conf.19pp, list(pairwise ~ Time|Subskill), adjust = "tukey")
```

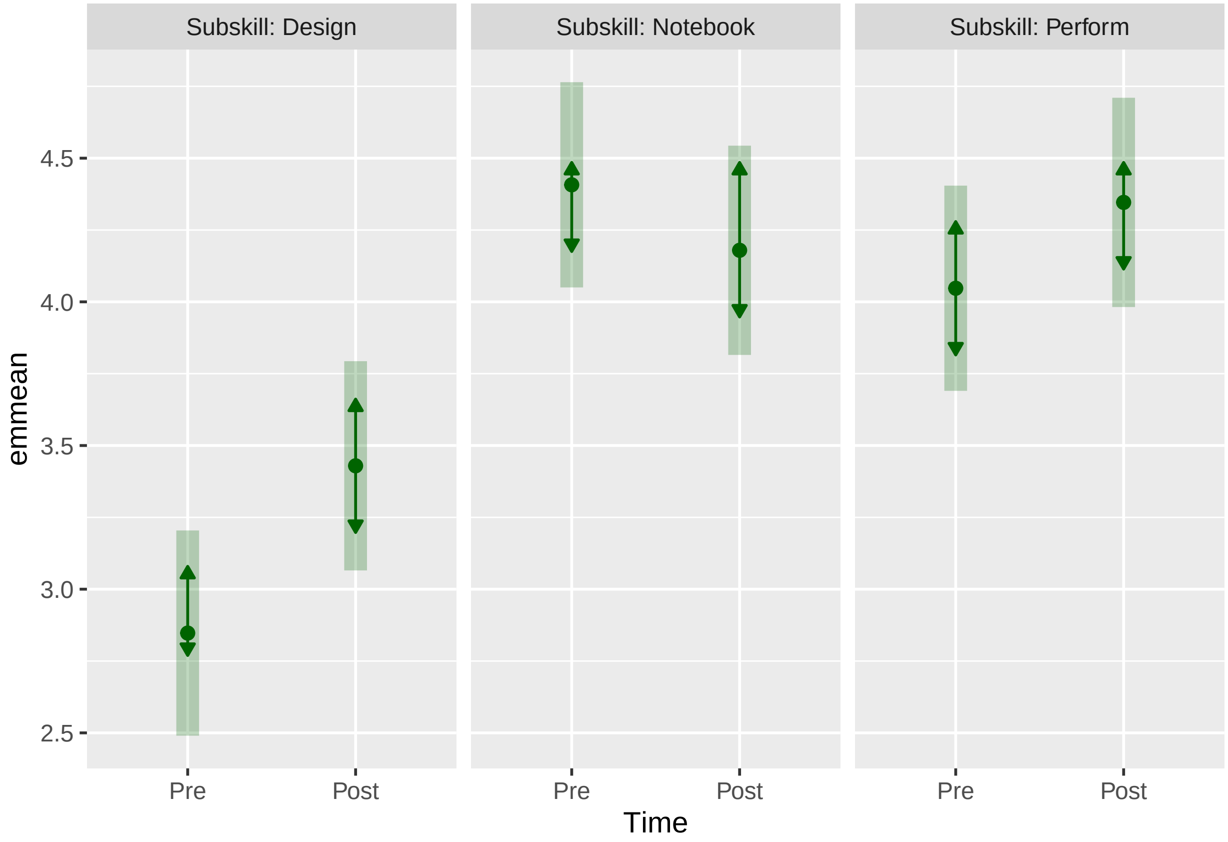
```
## Warning in model.frame.default(formula, data = data, ...): variable 'Time' is
## not a factor
```

```
confmeans19
```

```
## $`emmeans of Time | Subskill`
## Subskill = Design:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    2.85 0.175 32     2.49     3.20
##  Post   3.43 0.179 32     3.07     3.79
##
```

```
## Subskill = Notebook:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    4.41 0.175 32     4.05     4.76
##  Post   4.18 0.179 32     3.82     4.54
##
## Subskill = Perform:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    4.05 0.175 32     3.69     4.40
##  Post   4.35 0.179 32     3.98     4.71
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $`pairwise differences of Time | Subskill`
## Subskill = Design:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post   -0.582 0.234 109 -2.484  0.0145
##
## Subskill = Notebook:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post    0.228 0.234 109  0.972  0.3330
##
## Subskill = Perform:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post   -0.299 0.234 109 -1.275  0.2050
##
## Degrees-of-freedom method: containment
```

```
plot(confmeans19, by = "Subskill", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )
```



18.19 pre vs post data: Applicability Measures

```
#Pre and post comparison in 2018
app.18=subset(app, Experience == "Guided")
app.18=subset(app.18, Subskill != "Weakness")

app.18pp=lme(Score~Time*Subskill, random= ~1|ID, data=app.18, na.action=na.omit)

anova(app.18pp)
```

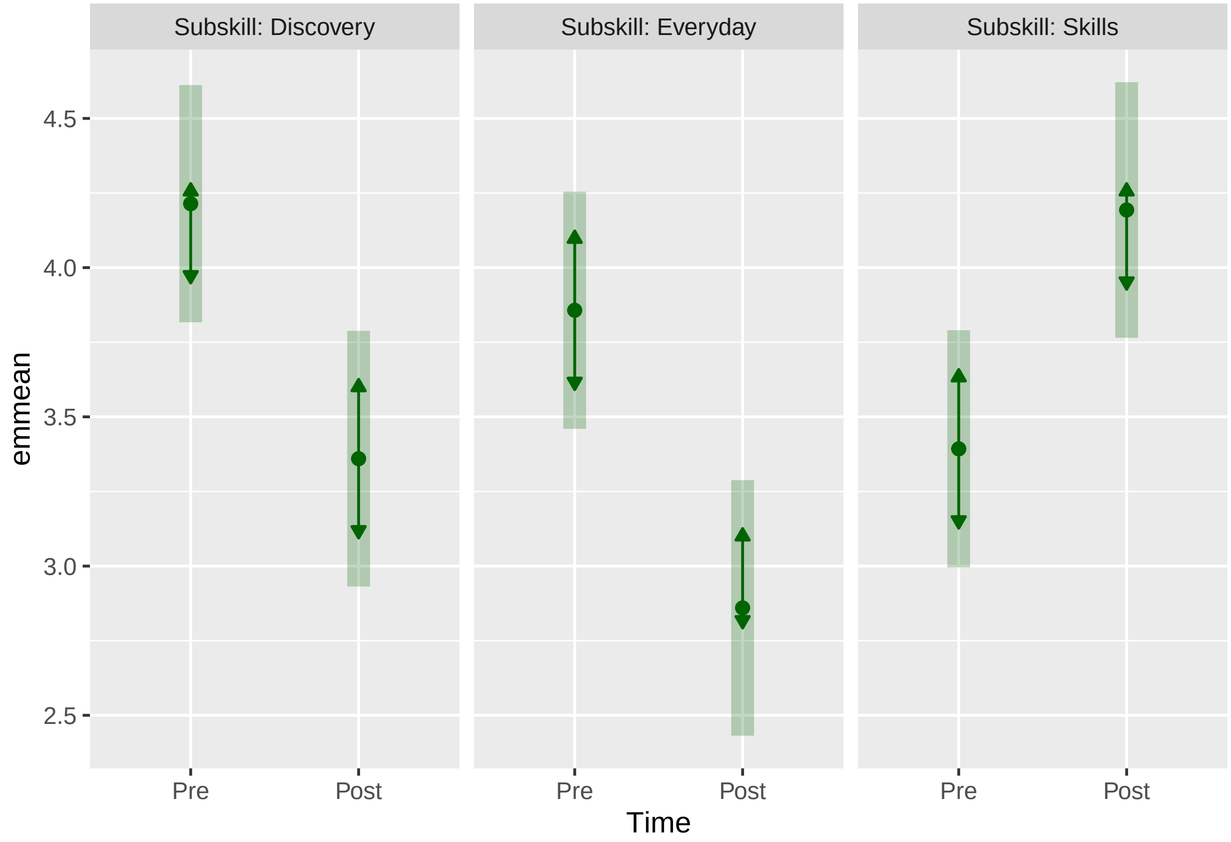
```
##               numDF denDF   F-value p-value
## (Intercept)       1   123 1288.0750  <.0001
## Time              1   123    5.0227  0.0268
## Subskill          2   123    2.9497  0.0561
## Time:Subskill     2   123   13.9036  <.0001
```

```
appmeans18=emmeans(app.18pp, list(pairwise ~ Time|Subskill), adjust = "tukey")
appmeans18
```

```
## $`emmeans of Time | Subskill`
## Subskill = Discovery:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    4.21 0.194 27     3.82     4.61
##  Post   3.36 0.209 27     2.93     3.79
##
```

```
## Subskill = Everyday:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    3.86 0.194 27     3.46     4.25
##  Post   2.86 0.209 27     2.43     3.29
##
## Subskill = Skills:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    3.39 0.194 27     3.00     3.79
##  Post   4.19 0.209 27     3.77     4.62
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $`pairwise differences of Time | Subskill`
## Subskill = Discovery:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post    0.854 0.269 123   3.177  0.0019
##
## Subskill = Everyday:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post    0.997 0.269 123   3.708  0.0003
##
## Subskill = Skills:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post   -0.800 0.269 123 -2.976  0.0035
##
## Degrees-of-freedom method: containment
```

```
plot(appmeans18, by = "Subskill", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )
```



```
#Pre and post comparison in 2019
app.19=subset(app, Experience == "Autonomous")
app.19=subset(app.19, Subskill != "Weakness")

app.19pp=lme(Score~Time*Subskill, random= ~1|ID, data=app.19, na.action=na.omit)

anova(app.19pp)
```

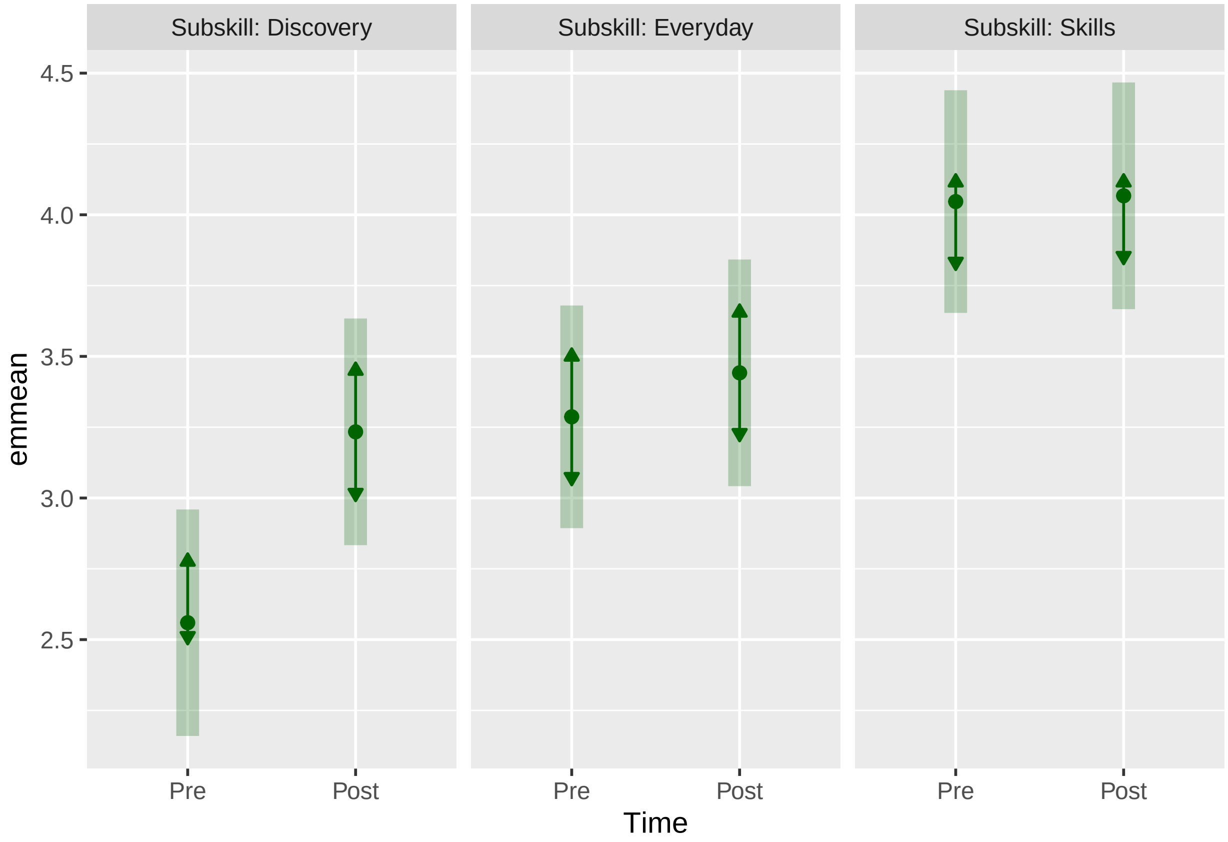
```
##                 numDF denDF  F-value p-value
## (Intercept)         1   108 838.6822  <.0001
## Time                1   108   3.1568  0.0784
## Subskill            2   108  24.8939  <.0001
## Time:Subskill       2   108   2.1561  0.1207
```

```
appmeans19=emmeans(app.19pp, list(pairwise ~ Time|Subskill), adjust = "tukey")
appmeans19
```

```
## $`emmeans of Time | Subskill`
## Subskill = Discovery:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    2.56 0.196 32     2.16     2.96
##  Post   3.23 0.196 32     2.83     3.63
##
## Subskill = Everyday:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    3.29 0.193 32     2.89     3.68
##  Post   3.44 0.196 32     3.04     3.84
```

```
##
## Subskill = Skills:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    4.05 0.193 32     3.65     4.44
##  Post   4.07 0.196 32     3.67     4.47
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $`pairwise differences of Time | Subskill`
## Subskill = Discovery:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post  -0.6739 0.246 108 -2.741  0.0072
##
## Subskill = Everyday:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post  -0.1553 0.243 108 -0.640  0.5237
##
## Subskill = Skills:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post  -0.0203 0.243 108 -0.084  0.9334
##
## Degrees-of-freedom method: containment
```

```
plot(appmeans19, by = "Subskill", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )
```



pre-post combined

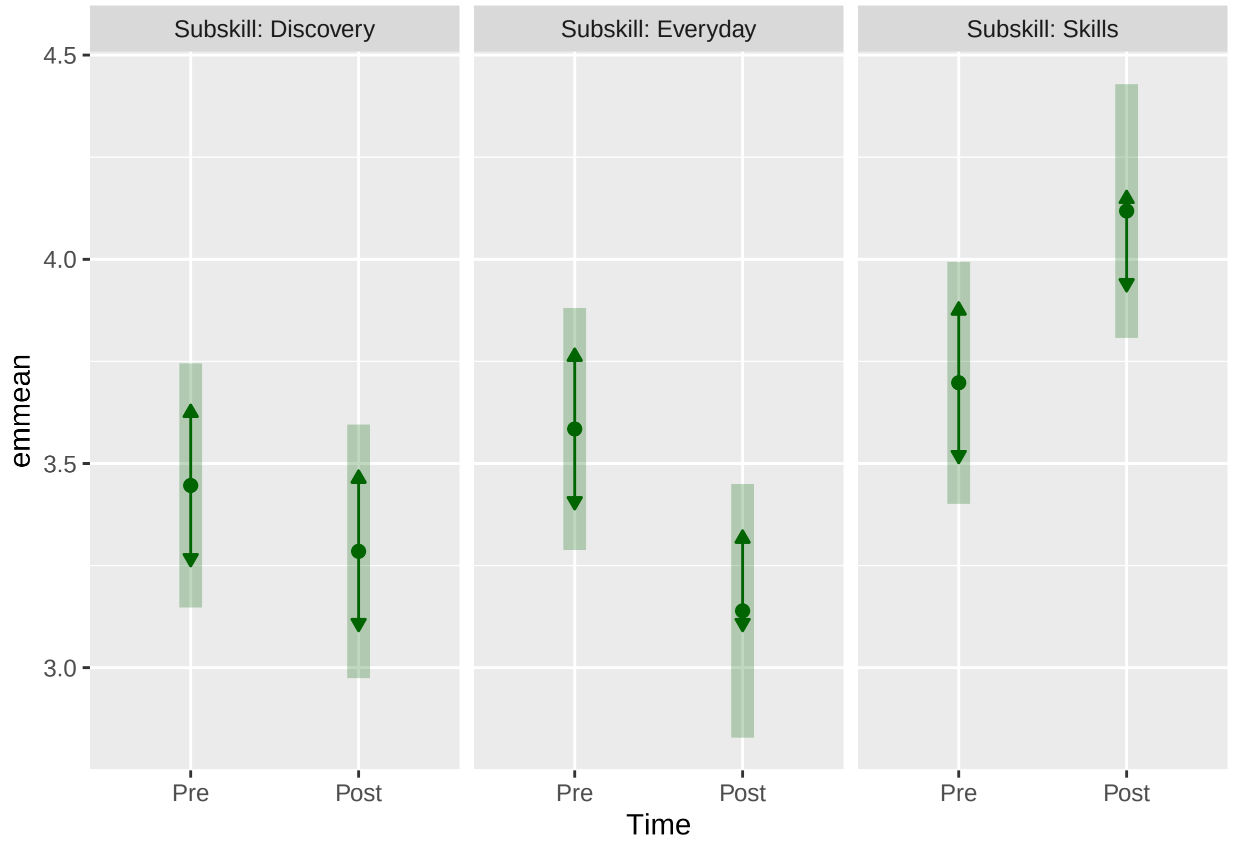
```
app.pp=lme(Score~Time*Subskill, random= ~1|ID, data=app, na.action=na.omit)

anova(app.pp)
```

```
##                 numDF denDF   F-value p-value
## (Intercept)         1   236 1893.6718  <.0001
## Time                1   236    0.2834  0.5950
## Subskill            2   236    9.5847  0.0001
## Time:Subskill       2   236    5.0453  0.0072
```

```
appmeans=emmeans(app.pp, list(pairwise ~ Time|Subskill), adjust = "tukey")
appmeans
```

```
## $`emmeans of Time | Subskill`
## Subskill = Discovery:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    3.45 0.149 60     3.15     3.74
##  Post   3.28 0.155 60     2.97     3.60
##
## Subskill = Everyday:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    3.58 0.148 60     3.29     3.88
##  Post   3.14 0.155 60     2.83     3.45
##
## Subskill = Skills:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    3.70 0.148 60     3.40     3.99
##  Post   4.12 0.155 60     3.81     4.43
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $`pairwise differences of Time | Subskill`
## Subskill = Discovery:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post    0.161 0.200 236   0.805  0.4217
##
## Subskill = Everyday:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post    0.445 0.199 236   2.235  0.0263
##
## Subskill = Skills:
##  contrast   estimate    SE  df t.ratio p.value
##  Pre - Post   -0.421 0.199 236  -2.110  0.0359
##
## Degrees-of-freedom method: containment
```

```
plot(appmeans, by = "Subskill", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )
```



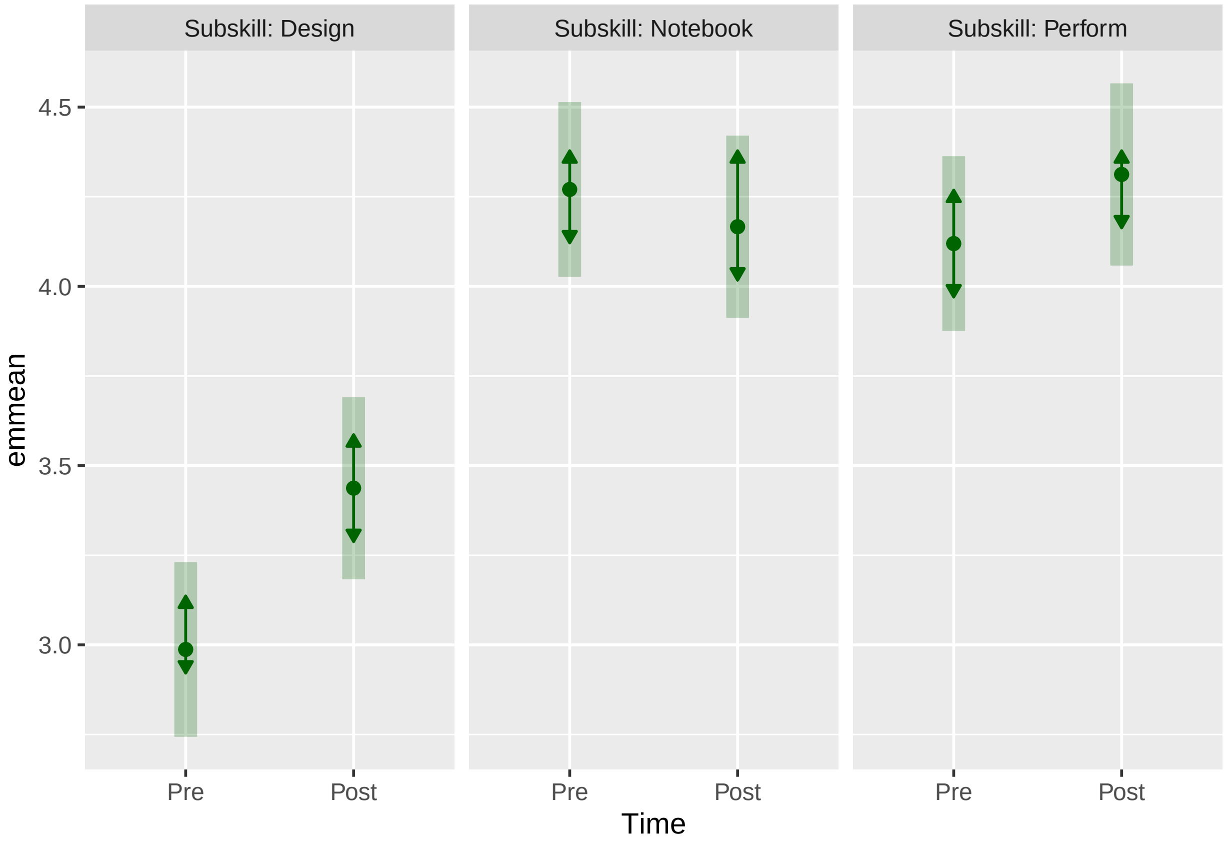
```
conf.w=subset(conf, Subskill != "Weakness")
con.pp=lme(Score~Time*Subskill, random= ~1|ID, data=conf.w, na.action=na.omit)

anova(con.pp)
```

```
##                 numDF denDF   F-value p-value
## (Intercept)         1   237 2472.7409  <.0001
## Time                1   237    3.8769  0.0501
## Subskill            2   237   62.6346  <.0001
## Time:Subskill       2   237    3.4950  0.0319
```

```
conmeans=emmeans(con.pp, list(pairwise ~ Time|Subskill), adjust = "tukey")
conmeans
```

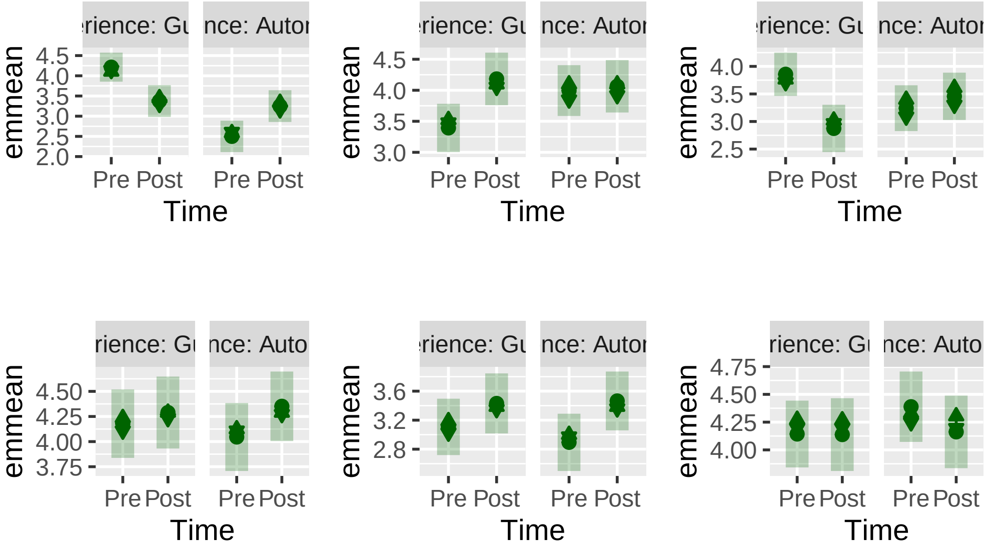
```
## $`emmeans of Time | Subskill`
## Subskill = Design:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    2.99 0.122 60     2.74     3.23
##  Post   3.44 0.127 60     3.18     3.69
##
## Subskill = Notebook:
##  Time emmean    SE df lower.CL upper.CL
##  Pre    4.27 0.122 60     4.03     4.51
##  Post   4.17 0.127 60     3.91     4.42
##
## Subskill = Perform:
##  Time emmean    SE df lower.CL upper.CL
##   Pre    4.12 0.122 60     3.88     4.36
##   Post   4.31 0.127 60     4.06     4.57
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $`pairwise differences of Time | Subskill`
## Subskill = Design:
##   contrast   estimate    SE  df t.ratio p.value
##   Pre - Post   -0.450 0.152 237 -2.968  0.0033
##
## Subskill = Notebook:
##   contrast   estimate    SE  df t.ratio p.value
##   Pre - Post    0.104 0.152 237  0.686  0.4934
##
## Subskill = Perform:
##   contrast   estimate    SE  df t.ratio p.value
##   Pre - Post   -0.193 0.152 237 -1.272  0.2046
##
## Degrees-of-freedom method: containment
```

```
plot(conmeans, by = "Subskill", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )
```



multipanel plot of pre-post

```r
des=subset(conf, Subskill == "Design")
des.lm=lme(Score~Time*Experience, random= ~1|ID, data=des, na.action=na.omit)
des.means=emmeans(des.lm, list(pairwise ~ Time|Experience), adjust = "tukey")
des.pl=plot(des.means, by = "Experience", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )

per=subset(conf, Subskill == "Perform")
per.lm=lme(Score~Time*Experience, random= ~1|ID, data=per, na.action=na.omit)
per.means=emmeans(per.lm, list(pairwise ~ Time|Experience), adjust = "tukey")
per.pl=plot(per.means, by = "Experience", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )

nb=subset(conf, Subskill == "Notebook")
nb.lm=lme(Score~Time*Experience, random= ~1|ID, data=nb, na.action=na.omit)
nb.means=emmeans(nb.lm, list(pairwise ~ Time|Experience), adjust = "tukey")
nb.pl=plot(nb.means, by = "Experience", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )

dis=subset(app, Subskill == "Discovery")
dis.lm=lme(Score~Time*Experience, random= ~1|ID, data=dis, na.action=na.omit)
dis.means=emmeans(dis.lm, list(pairwise ~ Time|Experience), adjust = "tukey")
dis.pl=plot(dis.means, by = "Experience", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )

sk=subset(app, Subskill == "Skills")
sk.lm=lme(Score~Time*Experience, random= ~1|ID, data=sk, na.action=na.omit)
sk.means=emmeans(sk.lm, list(pairwise ~ Time|Experience), adjust = "tukey")
sk.pl=plot(sk.means, by = "Experience", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )

ed=subset(app, Subskill == "Everyday")
ed.lm=lme(Score~Time*Experience, random= ~1|ID, data=ed, na.action=na.omit)
ed.means=emmeans(ed.lm, list(pairwise ~ Time|Experience), adjust = "tukey")
ed.pl=plot(ed.means, by = "Experience", comparisons = TRUE, horizontal = FALSE, color="darkgreen" )

p3=ggdraw() +
  draw_plot(ed.pl, x = 0.66, y = 0.5, width = 0.33, height = .4) +
  draw_plot(sk.pl, x = 0.33, y = 0.5, width = 0.33, height = .4) +
  draw_plot(dis.pl, x = 0, y = 0.5, width = 0.33, height = .4) +
  draw_plot(nb.pl, x = 0.66, y = 0, width = 0.33, height = .4)+
  draw_plot(des.pl, x = 0.33, y = 0, width = 0.33, height = 0.4) +
  draw_plot(per.pl, x = 0, y = 0, width = 0.33, height = 0.4) +

  theme(plot.margin=unit(c(15,15,15,15),"mm"))

ggsave(p3, file="pre.post.panel.png", width=12, height=8, dpi=600)

p3
```


rience: Gu
nce: Auton
emmean
4.5
4.0
3.5
3.0
2.5
2.0
Pre
Post
Time
rience: Gu
nce: Auton
emmean
4.5
4.0
3.5
3.0
Pre
Post
Time
rience: Gu
nce: Auton
emmean
4.0
3.5
3.0
2.5
Pre
Post
Time
rience: Gu
nce: Auto
emmean
4.50
4.25
4.00
3.75
Pre
Post
Time
rience: Gu
nce: Auton
emmean
3.6
3.2
2.8
Pre
Post
Time
rience: Gu
nce: Auto
emmean
4.75
4.50
4.25
4.00
Pre
Post
Time