Cheatsheet for the minimum stats/plotting needed for UG Psychology.

February 6, 2018

Pre-reqs

Working with data

Have a look at the data:

```
airquality %>% glimpse
```

```
## Observations: 153
## Variables: 6
## $ Ozone   <int> 41, 36, 12, 18, NA, 28, ...
## $ Solar.R <int> 190, 118, 149, 313, NA, ...
## $ Wind    <dbl> 7.4, 8.0, 12.6, 11.5, 14...
## $ Temp    <int> 67, 72, 74, 62, 56, 66, ...
## $ Month   <int> 5, 5, 5, 5, 5, 5, 5, 5, ...
## $ Day     <int> 1, 2, 3, 4, 5, 6, 7, 8, ...
```

```
airquality %>% head
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

Selecting columns, filtering rows and sorting

Select rows and or columns; sorting; show first 3:

```
gapminder::gapminder %>% filter(lifeExp > 80) %>%
    arrange(year) %>% select(year, country) %>%
    head(3)
```

```
## # A tibble: 3 x 2
##    year          country
##   <int>           <fctr>
## 1  1997            Japan
## 2  2002        Australia
## 3  2002 Hong Kong, China
```

Group by, and summarise

```
lme4::sleepstudy %>% group_by(Subject) %>% summarise(mean(Reaction)) %>%
    head(5)
```

```
## # A tibble: 5 x 2
##   Subject ‘mean(Reaction)‘
##    <fctr>            <dbl>
## 1     308         342.1338
## 2     309         215.2330
## 3     310         231.0013
## 4     330         303.2214
## 5     331         309.4361
```

Reshaping long to wide

Filter, select and spread wide:

```
gapminder::gapminder %>% filter(year > 2000) %>%
    select(year, country, lifeExp) %>% spread(year,
    lifeExp) %>% head(5)
```

```
## # A tibble: 5 x 3
##       country ‘2002‘ ‘2007‘
##        <fctr>  <dbl>  <dbl>
## 1 Afghanistan 42.129 43.828
## 2     Albania 75.651 76.423
## 3     Algeria 70.994 72.301
## 4      Angola 41.003 42.731
## 5   Argentina 74.340 75.320
```

Reshaping wide to long

Gather from wide to long form:

```
iris %>% gather(variable, value, -Species) %>%
    head
```

```
##   Species     variable value
## 1  setosa Sepal.Length   5.1
## 2  setosa Sepal.Length   4.9
## 3  setosa Sepal.Length   4.7
## 4  setosa Sepal.Length   4.6
## 5  setosa Sepal.Length   5.0
## 6  setosa Sepal.Length   5.4
```

Or using `data.table` and `melt`:

```
library(data.table)
iris %>% melt(id.var = "Species") %>% sample_n(10)
```

```
##         Species     variable value
## 382 versicolor Petal.Length   3.7
## 101  virginica Sepal.Length   6.3
## 514 versicolor  Petal.Width   1.4
## 25      setosa Sepal.Length   4.8
## 265  virginica  Sepal.Width   2.8
## 574  virginica  Petal.Width   1.8
## 250 versicolor  Sepal.Width   2.8
## 337     setosa Petal.Length   1.3
## 146  virginica Sepal.Length   6.7
## 180     setosa  Sepal.Width   3.2
```

Joining two separate datasets:

```
demographics <- data_frame(person = c(1, 2, 3),
    age = c(23, 25, 21))
responses <- data_frame(person = c(1, 1, 1, 2,
    2, 2), trial = c(1, 2, 3, 1, 2, 3), rt = c(230,
    245, 232, 343, 356, 374))

left_join(responses, demographics, by = "person")
```

```
## # A tibble: 6 x 4
##   person trial    rt   age
##    <dbl> <dbl> <dbl> <dbl>
## 1      1     1   230    23
## 2      1     2   245    23
## 3      1     3   232    23
## 4      2     1   343    25
## 5      2     2   356    25
## 6      2     3   374    25
```

Creating combinations for conditions for an experiment design

```
design <- expand.grid(condition = c("A", "B"),
    stimulus = LETTERS[1:3], participant = 1:20)

design %>% head(10)
```

```
##    condition stimulus participant
## 1          A        A           1
## 2          B        A           1
## 3          A        B           1
```

```
## 4          B        B           1
## 5          A        C           1
## 6          B        C           1
## 7          A        A           2
## 8          B        A           2
## 9          A        B           2
## 10         B        B           2
```

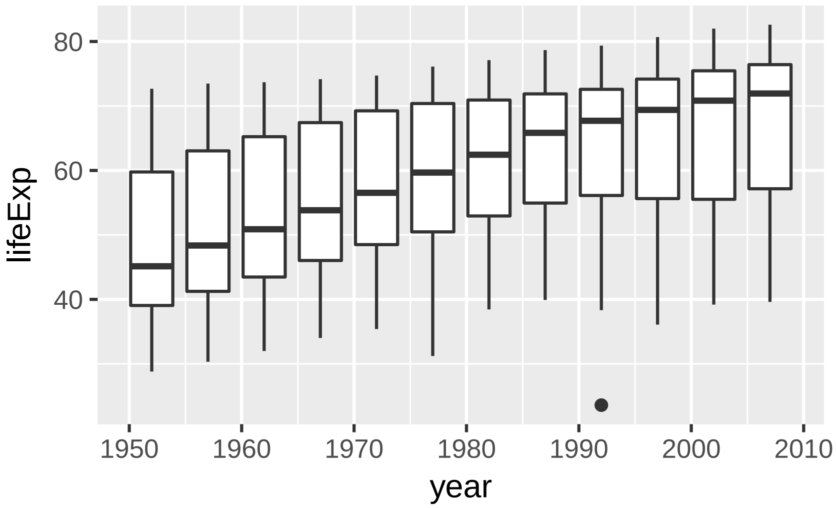
Randomising presentation within-participant:

```
design %>% mutate(i = runif(n())) %>% arrange(participant,
    i) %>% mutate(trial = row_number()) %>% select(participant,
    trial, everything(), -i) %>% head(10)
```

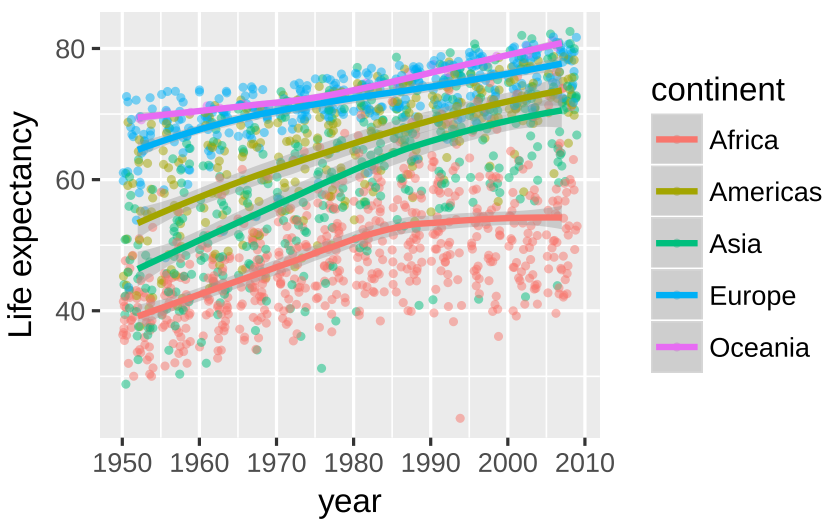
```
##    participant trial condition stimulus
## 1            1     1         B        B
## 2            1     2         B        C
## 3            1     3         A        A
## 4            1     4         A        C
## 5            1     5         A        B
## 6            1     6         B        A
## 7            2     7         A        A
## 8            2     8         B        B
## 9            2     9         B        C
## 10           2    10         A        B
```

Plotting

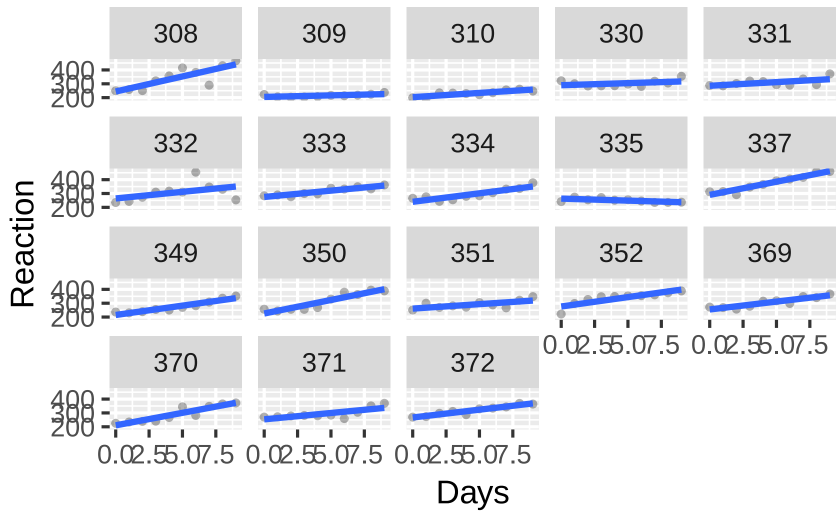
```
gapminder::gapminder %>% ggplot(aes(year, lifeExp,
    group = year)) + geom_boxplot()
```



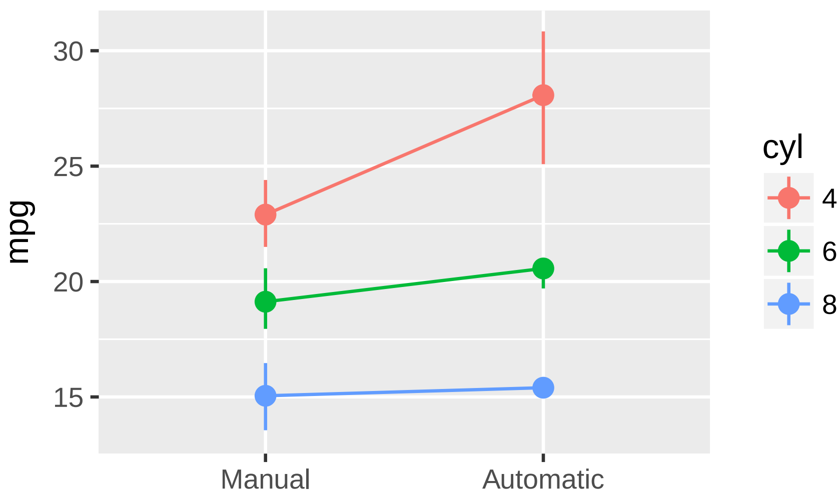

```
gapminder::gapminder %>% ggplot(aes(year, lifeExp,
    color = continent)) + geom_point(position = "jitter",
    alpha = 0.5, size = 0.75) + geom_smooth() +
    ylab("Life expectancy")
```




```
lme4::sleepstudy %>% ggplot(aes(Days, Reaction)) +
    geom_point(alpha = 0.3, size = 0.75) + geom_smooth(se = F,
    method = "lm") + facet_wrap(~Subject)
```



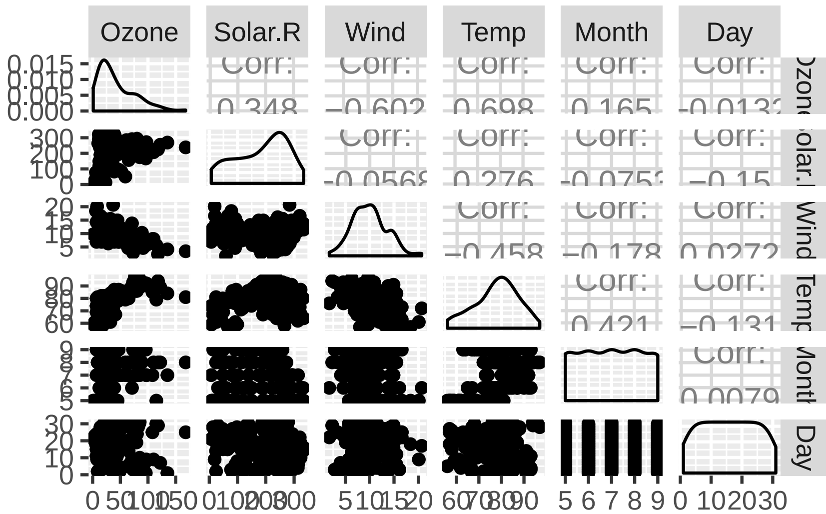

```
mtcars %>% mutate(am = factor(am, labels = c("Manual",
    "Automatic")), cyl = factor(cyl)) %>% ggplot(aes(am,
    mpg, color = cyl, group = cyl)) + stat_summary(geom = "pointrange",
    fun.data = mean_cl_boot) + stat_summary(geom = "line",
    fun.data = mean_cl_boot) + xlab("")
```



Correlation

As a plot:

```
GGally::ggpairs(airquality)
```



A table in APA format:

```
apaTables::apa.cor.table(airquality)
```

```
##
##
## Means, standard deviations, and correlations with confidence intervals
##
##
##   Variable   M      SD    1
##   1. Ozone   42.13  32.99
##
##   2. Solar.R 185.93 90.06 .35**
##                           [.17, .50]
##
##   3. Wind    9.96   3.52  -.60**
##                           [-.71, -.47]
##
##   4. Temp    77.88  9.47  .70**
##                           [.59, .78]
##
##   5. Month   6.99   1.42  .16
##                           [-.02, .34]
##
##   6. Day     15.80  8.86  -.01
##                           [-.20, .17]
##
```

```
##
##   2           3            4
##
##
##
##
##
##   -.06
##   [-.22, .11]
##
##   .28**       -.46**
##   [.12, .42]  [-.57, -.32]
##
##   -.08        -.18*        .42**
##   [-.23, .09] [-.33, -.02] [.28, .54]
##
##   -.15        .03          -.13
##   [-.31, .01] [-.13, .19]  [-.28, .03]
##
##   5
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##   -.01
##   [-.17, .15]
##
##
## Note. * indicates p < .05; ** indicates p < .01.
## M and SD are used to represent mean and standard deviation, respectively.
## Values in square brackets indicate the 95% confidence interval.
## The confidence interval is a plausible range of population correlations
## that could have caused the sample correlation (Cumming, 2014).
##
```

As a data_frame:

```
air.cor <- psych::corr.test(airquality)
air.cor$r %>% broom::tidy()
```

```
##   .rownames       Ozone     Solar.R
## 1     Ozone  1.00000000  0.34834169
## 2   Solar.R  0.34834169  1.00000000
## 3      Wind -0.60154653 -0.05679167
## 4      Temp  0.69836034  0.27584027
## 5     Month  0.16451931 -0.07530076
## 6       Day -0.01322565 -0.15027498
##          Wind       Temp        Month
## 1 -0.60154653  0.6983603  0.164519314
## 2 -0.05679167  0.2758403 -0.075300764
## 3  1.00000000 -0.4579879 -0.178292579
## 4 -0.45798788  1.0000000  0.420947252
## 5 -0.17829258  0.4209473  1.000000000
## 6  0.02718090 -0.1305932 -0.007961763
##            Day
## 1 -0.013225647
## 2 -0.150274979
## 3  0.027180903
## 4 -0.130593175
## 5 -0.007961763
## 6  1.000000000
```

Chi Sq

First make a table of counts with `xtabs`:

```
geartable <- xtabs(~gear + cyl, data = mtcars)
geartable
```

```
##     cyl
## gear  4  6  8
##    3  1  2 12
##    4  8  4  0
##    5  2  1  2
```

Then run the test on the table:

```
chisq.test(geartable)
```

```
##
##  Pearson's Chi-squared test
##
## data:  geartable
## X-squared = 18.036, df = 4, p-value =
## 0.001214
```

Linear models

Examples of simple regression; multiple regression with interaction; multiple regression with interaction of continuous by categorical predictors.

```
m1 <- lm(mpg ~ wt, data = mtcars)
m2 <- lm(mpg ~ wt * cyl, data = mtcars)
m3 <- lm(mpg ~ wt * factor(cyl), data = mtcars)
```

Coefficient summaries from models

```
summary(m1)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value
## (Intercept)  37.2851     1.8776  19.858
## wt           -5.3445     0.5591  -9.559
##             Pr(>|t|)
## (Intercept)  < 2e-16 ***
## wt          1.29e-10 ***
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

Extracting model coefficients as a dataframe:

```
m2 %>% broom::tidy(conf.int = T) %>% pander
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	54.31	6.128	8.863	1.289e-09	41.76	66.86
wt	-8.656	2.32	-3.731	0.000861	-13.41	-3.903

term	estimate	std.error	statistic	p.value	conf.low	conf.high
cyl	-3.803	1.005	-3.784	0.0007472	-5.862	-1.745
wt:cyl	0.8084	0.3273	2.47	0.01988	0.1379	1.479

Extracting model fit details too:

```
m2 %>% broom::glance() %>% pander
```

Table 2: Table continues below

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik
0.8606	0.8457	2.368	57.62	4.231e-12	4	-70.85

AIC	BIC	deviance	df.residual
151.7	159	157	28

Anova tables

```
car::Anova(m3, type = 3)
```

```
## Anova Table (Type III tests)
##
## Response: mpg
##                 Sum Sq Df  F value    Pr(>F)
## (Intercept)     920.34  1 153.4987 2.058e-12
## wt              103.45  1  17.2537 0.0003128
## factor(cyl)      64.48  2   5.3769 0.0111106
## wt:factor(cyl)  27.17  2   2.2658 0.1238570
## Residuals       155.89 26
##
## (Intercept)     ***
## wt              ***
## factor(cyl)     *
## wt:factor(cyl)
## Residuals
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
```

Format in APA format:

```
apaTables::apa.aov.table(m3)
```

```
##
##
## ANOVA results using mpg as the dependent variable
##
##
##         Predictor     SS df     MS      F
##       (Intercept) 920.34  1 920.34 153.50
##                wt 103.45  1 103.45  17.25
##       factor(cyl)  64.48  2  32.24   5.38
##  wt x factor(cyl)  27.17  2  13.59   2.27
##             Error 155.89 26   6.00
##     p partial_eta2 CI_90_partial_eta2
##  .000
##  .000          .40         [.15, .56]
##  .011          .29         [.05, .46]
##  .124          .15         [.00, .31]
##
##
## Note: Values in square brackets indicate the bounds of the 90% confidence interval for partial eta-squa
```

Post-hoc tests

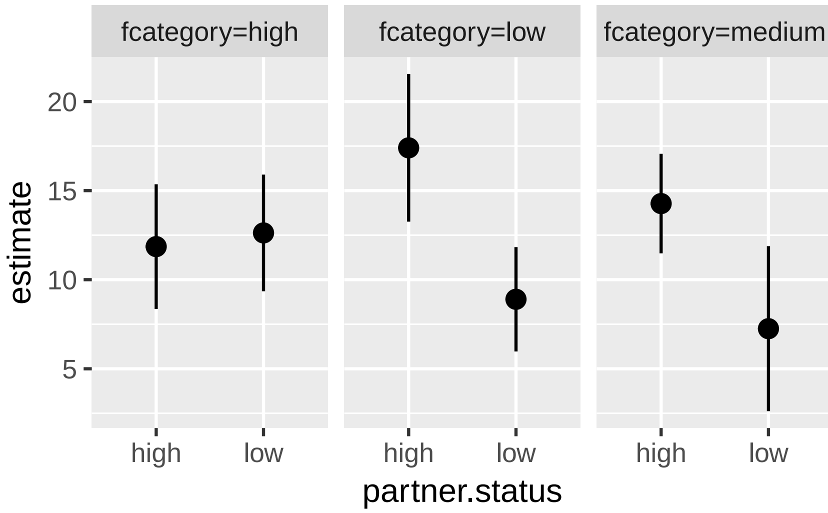
Run a suitable model:

```
m4 <- lm(conformity ~ fcategory * partner.status,
    data = car::Moore)
car::Anova(m4)
```

```
## Anova Table (Type II tests)
##
## Response: conformity
##                          Sum Sq Df F value
## fcategory                 11.61  2  0.2770
## partner.status           212.21  1 10.1207
## fcategory:partner.status 175.49  2  4.1846
## Residuals                817.76 39
##                            Pr(>F)
## fcategory                0.759564
## partner.status           0.002874 **
## fcategory:partner.status 0.022572 *
## Residuals
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
```

And plot the marginal means:

```
lsmeans::lsmeans(m4, ~partner.status * fcategory) %>%
    broom::tidy() %>% ggplot(aes(partner.status,
    estimate, ymin = conf.low, ymax = conf.high)) +
    geom_pointrange() + facet_wrap(~paste0("fcategory=",
    fcategory))
```



And extract the pairwise test statistics with FDR correction:

```
pairs(lsmeans::lsmeans(m4, ~partner.status * fcategory),
    adjust = "fdr") %>% broom::tidy() %>% pander(caption = "Pairwise tests, with FDR correction.")
```

Table 4: Pairwise tests, with FDR correction.

level1	level2	estimate	std.error	df	statistic	p.value
high,high	low,high	-0.7679	2.37	39	-0.324	0.7477
high,high	high,low	-5.543	2.681	39	-2.067	0.1362
high,high	low,low	2.957	2.257	39	1.31	0.2904
high,high	high,medium	-2.416	2.214	39	-1.091	0.3524
high,high	low,medium	4.607	2.87	39	1.605	0.1942
low,high	high,low	-4.775	2.61	39	-1.829	0.1608
low,high	low,low	3.725	2.172	39	1.715	0.1768
low,high	high,medium	-1.648	2.128	39	-0.7744	0.5116
low,high	low,medium	5.375	2.804	39	1.917	0.1565
high,low	low,low	8.5	2.508	39	3.389	0.01536
high,low	high,medium	3.127	2.47	39	1.266	0.2904
high,low	low,medium	10.15	3.072	39	3.304	0.01536
low,low	high,medium	-5.373	2.001	39	-2.685	0.04597
low,low	low,medium	1.65	2.709	39	0.6091	0.585
high,medium	low,medium	7.023	2.674	39	2.627	0.04597

Repeated measures Anova/Mixed Models

These models are (roughly) equivalent ways of doing the similar things:

```
data(obk.long, package = "afex")

rm1 <- afex::aov_4(value ~ treatment * gender +
    (phase * hour | id), data = obk.long, observed = "gender")

rm2 <- lmerTest::lmer(value ~ treatment * gender *
    hour * phase + (1 | id), data = obk.long)
```

The Anova tables for each model:

```
anova(rm1)
```

```
## Anova Table (Type 3 tests)
##
## Response: value
##                              num Df den Df
## treatment                    2.0000 10.000
## gender                       1.0000 10.000
## treatment:gender             2.0000 10.000
## phase                        1.5991 15.991
## treatment:phase              3.1981 15.991
## gender:phase                 1.5991 15.991
## treatment:gender:phase       3.1981 15.991
## hour                         1.8411 18.411
## treatment:hour               3.6823 18.411
## gender:hour                  1.8411 18.411
## treatment:gender:hour        3.6823 18.411
## phase:hour                   3.5960 35.960
## treatment:phase:hour         7.1920 35.960
## gender:phase:hour            3.5960 35.960
## treatment:gender:phase:hour  7.1920 35.960
##                                  MSE       F
## treatment                    22.8056  3.9405
## gender                       22.8056  3.6591
## treatment:gender             22.8056  2.8555
## phase                         5.0203 16.1329
## treatment:phase               5.0203  4.8510
## gender:phase                  5.0203  0.2828
## treatment:gender:phase        5.0203  0.6366
## hour                          3.3947 16.6857
## treatment:hour                3.3947  0.0933
```

```
## gender:hour                 3.3947  0.4503
## treatment:gender:hour       3.3947  0.6204
## phase:hour                  2.6743  1.1799
## treatment:phase:hour        2.6743  0.3453
## gender:phase:hour           2.6743  0.9313
## treatment:gender:phase:hour  2.6743  0.7359
##                                 ges
## treatment                  0.277906
## gender                     0.151601
## treatment:gender           0.218071
## phase                      0.217115
## treatment:phase            0.142939
## gender:phase               0.004838
## treatment:gender:phase     0.021418
## hour                       0.182545
## treatment:hour             0.002492
## gender:hour                0.005990
## treatment:gender:hour      0.016336
## phase:hour                 0.023721
## treatment:phase:hour       0.014021
## gender:phase:hour          0.018817
## treatment:gender:phase:hour 0.029418
##                               Pr(>F)
## treatment                  0.0547069 .
## gender                     0.0848003 .
## treatment:gender           0.1044692
## phase                      0.0002814 ***
## treatment:phase            0.0126909 *
## gender:phase               0.7089599
## treatment:gender:phase     0.6116209
## hour                       9.763e-05 ***
## treatment:hour             0.9786227
## gender:hour                0.6284344
## treatment:gender:hour      0.6413625
## phase:hour                 0.3345212
## treatment:phase:hour       0.9303725
## gender:phase:hour          0.4490777
## treatment:gender:phase:hour 0.6463449
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
```

```
lmerTest::anova(rm2)
```

```
## Analysis of Variance Table of type III  with  Satterthwaite
```

```
## approximation for degrees of freedom
##                              Sum Sq Mean Sq
## treatment                    13.451   6.725
## gender                        6.245   6.245
## hour                        104.285  26.071
## phase                       129.511  64.756
## treatment:gender              9.747   4.874
## treatment:hour                1.167   0.146
## gender:hour                   2.814   0.704
## treatment:phase              77.885  19.471
## gender:phase                  2.270   1.135
## hour:phase                   11.347   1.418
## treatment:gender:hour         7.755   0.969
## treatment:gender:phase       10.221   2.555
## treatment:hour:phase          6.641   0.415
## gender:hour:phase             8.956   1.119
## treatment:gender:hour:phase  14.155   0.885
##                             NumDF DenDF
## treatment                       2    10
## gender                          1    10
## hour                            4   140
## phase                           2   140
## treatment:gender                2    10
## treatment:hour                  8   140
## gender:hour                     4   140
## treatment:phase                 4   140
## gender:phase                    2   140
## hour:phase                      8   140
## treatment:gender:hour           8   140
## treatment:gender:phase          4   140
## treatment:hour:phase           16   140
## gender:hour:phase               8   140
## treatment:gender:hour:phase    16   140
##                             F.value
## treatment                     3.940
## gender                        3.659
## hour                         15.275
## phase                        37.941
## treatment:gender              2.855
## treatment:hour                0.085
## gender:hour                   0.412
## treatment:phase              11.408
## gender:phase                  0.665
## hour:phase                    0.831
```

```
## treatment:gender:hour         0.568
## treatment:gender:phase        1.497
## treatment:hour:phase          0.243
## gender:hour:phase             0.656
## treatment:gender:hour:phase   0.518
##                                Pr(>F)
## treatment                     0.05471 .
## gender                        0.08480 .
## hour                        2.175e-10 ***
## phase                       6.817e-14 ***
## treatment:gender              0.10447
## treatment:hour                0.99954
## gender:hour                   0.79962
## treatment:phase             4.822e-08 ***
## gender:phase                  0.51587
## hour:phase                    0.57670
## treatment:gender:hour         0.80271
## treatment:gender:phase        0.20631
## treatment:hour:phase          0.99886
## gender:hour:phase             0.72937
## treatment:gender:hour:phase   0.93420
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
```

Extract means/contrasts for groups:

```
lsmeans::lsmeans(rm1, ~treatment)
```

```
##  treatment   lsmean        SE df lower.CL
##  control   4.222222 0.5501512 10 2.996409
##  A         6.250000 0.5690190 10 4.982147
##  B         6.027778 0.5205689 10 4.867878
##  upper.CL
##  5.448035
##  7.517853
##  7.187677
##
## Results are averaged over the levels of: gender, phase, hour
## Confidence level used: 0.95
```

```
pairs(lsmeans::lsmeans(rm1, ~treatment))
```

```
##  contrast      estimate        SE df t.ratio
##  control - A -2.0277778 0.8347673 10  -2.429
##  control - B -1.8055556 0.7338014 10  -2.461
```

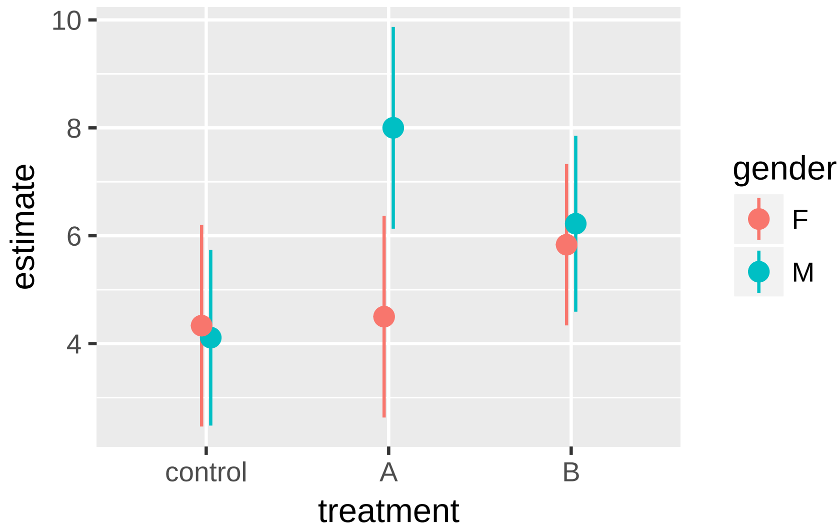
```
##  A - B        0.2222222 0.7757662 10   0.286
##  p.value
##   0.0830
##   0.0789
##   0.9560
##
## Results are averaged over the levels of: gender, phase, hour
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```
lmerTest::difflsmeans(rm2, "treatment")
```

```
## Differences of LSMEANS:
##                       Estimate
## treatment control - A     -2.0
## treatment control - B     -1.8
## treatment A - B            0.2
##                       Standard Error   DF
## treatment control - A          0.835 10.0
## treatment control - B          0.734 10.0
## treatment A - B                0.776 10.0
##                       t-value Lower CI
## treatment control - A   -2.43    -3.89
## treatment control - B   -2.46    -3.44
## treatment A - B          0.29    -1.51
##                       Upper CI p-value
## treatment control - A   -0.168    0.04 *
## treatment control - B   -0.171    0.03 *
## treatment A - B          1.951    0.78
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
```

Plotting the results of the model:

```
lsmeans::lsmeans(rm1, ~treatment * gender) %>%
    broom::tidy() %>% ggplot(aes(treatment, estimate,
    ymin = conf.low, ymax = conf.high, color = gender,
    group = gender)) + geom_pointrange(position = position_dodge(width = 0.1))
```



Within a mixed-model framework, we can also allow for varying effects of phase/hour by participant:

```
rm3 <- lmerTest::lmer(value ~ treatment * gender *
    hour * phase + (phase + hour | id), data = obk.long)
lmerTest::anova(rm3)
```

```
## Analysis of Variance Table of type III  with  Satterthwaite
## approximation for degrees of freedom
##                               Sum Sq Mean Sq
## treatment                      7.746  3.8732
## gender                         3.597  3.5966
## hour                          76.760 19.1900
## phase                         40.782 20.3910
## treatment:gender               5.613  2.8067
## treatment:hour                 1.064  0.1330
## gender:hour                    2.653  0.6632
## treatment:phase               20.666  5.1665
## gender:phase                   0.663  0.3317
## hour:phase                    11.347  1.4183
## treatment:gender:hour          3.425  0.4282
## treatment:gender:phase         4.061  1.0152
## treatment:hour:phase           6.641  0.4151
## gender:hour:phase              8.956  1.1195
## treatment:gender:hour:phase   14.155  0.8847
##                               NumDF   DenDF
## treatment                         2  10.000
## gender                            1  10.000
## hour                              4  19.075
## phase                             2  10.311
## treatment:gender                  2  10.000
```

```
## treatment:hour                   8  19.075
## gender:hour                      4  19.075
## treatment:phase                  4  10.311
## gender:phase                     2  10.311
## hour:phase                       8 110.000
## treatment:gender:hour            8  19.075
## treatment:gender:phase           4  10.311
## treatment:hour:phase            16 110.000
## gender:hour:phase                8 110.000
## treatment:gender:hour:phase     16 110.000
##                             F.value
## treatment                    3.9402
## gender                       3.6589
## hour                        19.5223
## phase                       20.7441
## treatment:gender             2.8553
## treatment:hour               0.1353
## gender:hour                  0.6747
## treatment:phase              5.2560
## gender:phase                 0.3375
## hour:phase                   1.4429
## treatment:gender:hour        0.4356
## treatment:gender:phase       1.0328
## treatment:hour:phase         0.4223
## gender:hour:phase            1.1389
## treatment:gender:hour:phase  0.9000
##                                Pr(>F)
## treatment                   0.0547148 .
## gender                      0.0848091 .
## hour                        1.564e-06 ***
## phase                       0.0002432 ***
## treatment:gender            0.1044818
## treatment:hour              0.9965256
## gender:hour                 0.6177058
## treatment:phase             0.0144727 *
## gender:phase                0.7211679
## hour:phase                  0.1867847
## treatment:gender:hour       0.8849641
## treatment:gender:phase      0.4356394
## treatment:hour:phase        0.9739000
## gender:hour:phase           0.3432511
## treatment:gender:hour:phase 0.5709270
## ---
## Signif. codes:
```

```
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
```

And we can test whether the random effects are different to zero:

```
lmerTest::rand(rm3)
```

```
## Analysis of Random effects Table:
##          Chi.sq Chi.DF p.value
## phase:id   30.5     13   0.004 **
## hour:id    20.8     22   0.532
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
```