UNM04

2023-07-03

The important columns that need to be taken in all data files are "participant" and "expName".

Training data start in row 6 of excel, 5 if you don't count the header, and finish in row 85, 84 if you don't count the header. The important columns here are: "cue_img", (what images are each cue, basically the randomization of the images for that participant) "cue_order", "out_order", "cue_o_mouse.time", (basically the reaction time) "cue_o_mouse.clicked_name", (which outcome they have clicked) "correct_answer", (1 if they clicked the outcome programmed, 0 if they clicked the other) "training_trials.thisN", (trial number) "cue1", "cue2", "outcome"

Test data start in row 86 (85 no header), and finish in row 93 (92 no header). The important columns here are: "test_mouse.time", "test_mouse.clicked_name", "slider.response" , "slider.rt", "test.thisTrialN", (trial number) "target"

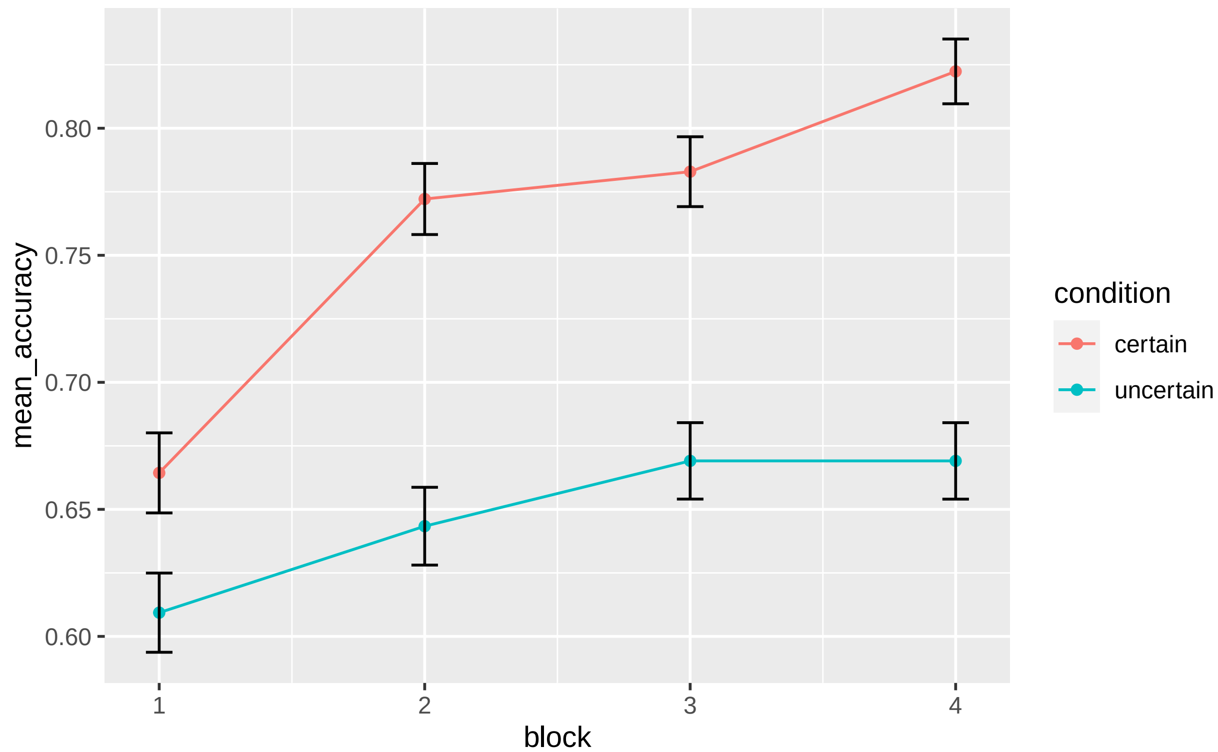
```
training$prob_response[training$RT > 10] <- NA
training$RT[training$RT > 10] <- NA #substitute more than 10 secs for NA
#Plot Training accuracy
MA_training <- training %>%
  group_by(condition, block) %>%
  summarise(mean_accuracy = mean(prob_response, na.rm = TRUE),
            sd_accuracy = sd(prob_response, na.rm = TRUE)/sqrt(length(prob_response)))
```

```
## `summarise()` has grouped output by 'condition'. You can override using the
## `.groups` argument.
```

```
ggplot(MA_training, mapping = aes(x = block, y = mean_accuracy, color = condition)) +
  geom_point() +
  geom_line() +
  geom_errorbar(aes(x= block, y = mean_accuracy, ymin = mean_accuracy-sd_accuracy, ymax = mean_accuracy+
  labs(title = "Figure 1", subtitle = "Mean corrected accuracy for the 4 blocks of the training phase")
```

Figure 1

Mean corrected accuracy for the 4 blocks of the training phase




```
#some t test to check that responding is significantly higher than chance
mean_training <- training %>%
  group_by(pNum) %>%
   summarise(mean_response = mean(prob_response, na.rm = TRUE))
t.test(mean_training, mu = .5, alternative = "greater")
```

```
##
##  One Sample t-test
##
## data:  mean_training
## t = 10.665, df = 187, p-value < 2.2e-16
## alternative hypothesis: true mean is greater than 0.5
## 95 percent confidence interval:
##  20.44277      Inf
## sample estimates:
## mean of x
##  24.10067
```

```
mean_cert_training <- filter(training, condition == "certain") %>%
  group_by(pNum) %>%
   summarise(mean_response = mean(prob_response, na.rm = TRUE))
t.test(mean_cert_training, mu = .5, alternative = "greater")
```

```
##
##  One Sample t-test
##
## data:  mean_cert_training
```

```
## t = 7.7724, df = 89, p-value = 6.35e-12
## alternative hypothesis: true mean is greater than 0.5
## 95 percent confidence interval:
##  20.35595      Inf
## sample estimates:
## mean of x
##  25.75731
```

```
mean_uncert_training <- filter(training, condition == "uncertain") %>%
  group_by(pNum) %>%
   summarise(mean_response = mean(prob_response, na.rm = TRUE))
t.test(mean_uncert_training, mu = .5, alternative = "greater")
```

```
##
##  One Sample t-test
##
## data:  mean_uncert_training
## t = 7.2947, df = 97, p-value = 4.094e-11
## alternative hypothesis: true mean is greater than 0.5
## 95 percent confidence interval:
##  17.55272      Inf
## sample estimates:
## mean of x
##  22.57927
```

```
#ANOVA
prob_resp <- training %>%
  group_by (pNum, block, condition) %>%
  summarise(mean_response = mean(prob_response, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'pNum', 'block'. You can override using the
## `.groups` argument.
```

```
prob_resp$block <- factor(prob_resp$block)
prob_resp$condition <- factor(prob_resp$condition)
prob_resp$pNum <- factor(prob_resp$pNum)
ANOVA_prob_resp <- aov_car(formula = mean_response ~ condition + Error(pNum/block), data = prob_resp)
```

```
## Contrasts set to contr.sum for the following variables: condition
```

```
print(ANOVA_prob_resp)
```

```
## Anova Table (Type 3 tests)
##
## Response: mean_response
##            Effect           df  MSE         F  ges p.value
## 1       condition        1, 92 0.10 12.04 *** .083   <.001
## 2           block 2.59, 238.34 0.02 15.17 *** .048   <.001
## 3 condition:block 2.59, 238.34 0.02    3.20 * .010    .030
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
##
## Sphericity correction method: GG
```

```
bay_ANOVA_prob_resp <- anovaBF(formula = mean_response ~ block*condition + pNum,
        data = data.frame(prob_resp),
        whichRandom = "pNum")
```

```r
print(bay_ANOVA_prob_resp)
```

```
## Bayes factor analysis
## --------------
## [1] block + pNum                                  : 992363.2 ±1.54%
## [2] condition + pNum                              : 37.47994 ±1.25%
## [3] block + condition + pNum                      : 35304957 ±1.25%
## [4] block + condition + block:condition + pNum : 45574922 ±1.48%
##
## Against denominator:
##   mean_response ~ pNum
## ---
## Bayes factor type: BFlinearModel, JZS
```

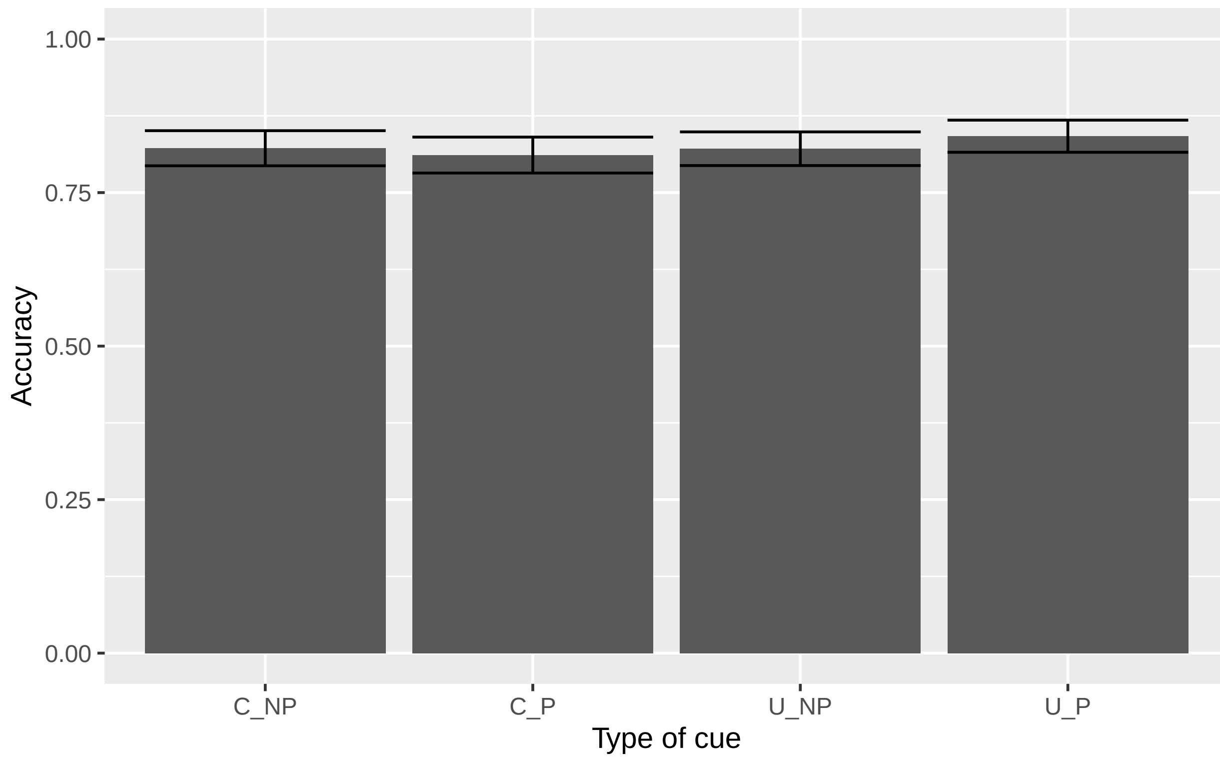
```r
bay_ANOVA_prob_resp[4]/bay_ANOVA_prob_resp[3]
```

```
## Bayes factor analysis
## --------------
## [1] block + condition + block:condition + pNum : 1.290893 ±1.94%
##
## Against denominator:
##   mean_response ~ block + condition + pNum
## ---
## Bayes factor type: BFlinearModel, JZS
```

```r
#plot test accuracy
m_acc_test <- test %>%
  group_by(cue_type) %>%
  summarise(mean_acc = mean(acc, na.rm = TRUE),
            sd_acc = sd(acc, na.rm = TRUE)/sqrt(length(acc)))
ggplot(data = m_acc_test) +
  geom_col(mapping = aes(x = cue_type, y = mean_acc)) +
  geom_errorbar(aes(x = cue_type, y= mean_acc, ymin = mean_acc - sd_acc, ymax = mean_acc + sd_acc)) +
  coord_cartesian(ylim = c(0, 1))+
  scale_x_discrete (name = "Type of cue") +
  scale_y_continuous(name = "Accuracy") +
  labs(title = "Figure 2", subtitle = "Mean accuracy for each type of cue in test phase")
```

Figure 2

Mean accuracy for each type of cue in test phase




```
#ANOVA accuracy
acc_test <- test %>%
  group_by (pNum, condition, predictiveness) %>%
  summarise(acc = mean(acc, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'pNum', 'condition'. You can override using
## the `.groups` argument.
```

```
acc_test$predictiveness <- factor(acc_test$predictiveness)
acc_test$condition <- factor(acc_test$condition)
acc_test$pNum <- factor(acc_test$pNum)
ANOVA_acc_test <- aov_car(formula = acc ~ condition + Error(pNum*predictiveness), data = acc_test)
```

```
## Contrasts set to contr.sum for the following variables: condition
```

```
print(ANOVA_acc_test)
```

```
## Anova Table (Type 3 tests)
##
## Response: acc
##                     Effect    df  MSE    F   ges p.value
## 1                condition 1, 92 0.07 0.15  .001    .698
## 2           predictiveness 1, 92 0.04 0.03 <.001    .872
## 3 condition:predictiveness 1, 92 0.04 0.30  .001    .584
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```r
bay_ANOVA_acc_test <- anovaBF(formula = acc ~ condition*predictiveness + pNum,
        data = data.frame(acc_test),
        whichRandom = "pNum")
print(bay_ANOVA_acc_test)
```

```
## Bayes factor analysis
## --------------
## [1] condition + pNum                                              : 0.2263669   ±1.14%
## [2] predictiveness + pNum                                         : 0.1558339   ±0.91%
## [3] condition + predictiveness + pNum                             : 0.03519834  ±1.75%
## [4] condition + predictiveness + condition:predictiveness + pNum : 0.008330032 ±1.59%
##
## Against denominator:
##   acc ~ pNum
## ---
## Bayes factor type: BFlinearModel, JZS
```

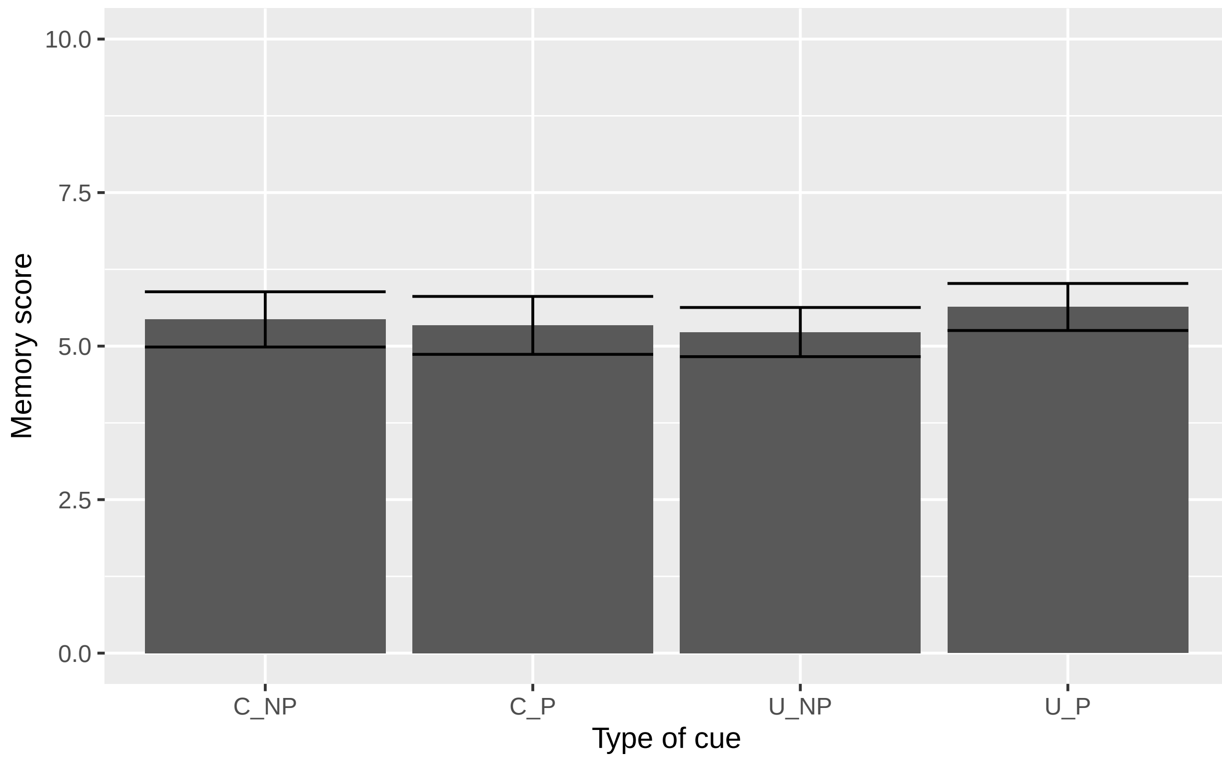
```r
bay_ANOVA_acc_test[4]/bay_ANOVA_acc_test[3]
```

```
## Bayes factor analysis
## --------------
## [1] condition + predictiveness + condition:predictiveness + pNum : 0.2366598 ±2.36%
##
## Against denominator:
##   acc ~ condition + predictiveness + pNum
## ---
## Bayes factor type: BFlinearModel, JZS
```

```r
#plot test mem_score
m_mem_test <- test %>%
  group_by(cue_type) %>%
  summarise(mean_mem_score = mean(mem_score, na.rm = TRUE),
            sd_mem_score = sd(mem_score, na.rm = TRUE)/sqrt(length(mem_score)))
ggplot(data = m_mem_test) +
  geom_col(mapping = aes(x = cue_type, y = mean_mem_score)) +
  geom_errorbar(aes(x = cue_type, y= mean_mem_score, ymin = mean_mem_score - sd_mem_score, ymax = mean_m
  coord_cartesian(ylim = c(0, 10))+
  scale_x_discrete (name = "Type of cue") +
  scale_y_continuous(name = "Memory score") +
  labs(title = "Figure 3", subtitle = "Mean memory score for each type of cue in test phase")
```

Figure 3

Mean memory score for each type of cue in test phase




```
#ANOVA mem_score
mem_score_test <- test %>%
  group_by (pNum, condition, predictiveness) %>%
  summarise(mem_score = mean(mem_score, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'pNum', 'condition'. You can override using
## the `.groups` argument.
```

```
mem_score_test$predictiveness <- factor(mem_score_test$predictiveness)
mem_score_test$condition <- factor(mem_score_test$condition)
mem_score_test$pNum <- factor(mem_score_test$pNum)
ANOVA_mem_score_test <- aov_car(formula = mem_score ~ condition + Error(pNum*predictiveness), data = mem
```

```
## Contrasts set to contr.sum for the following variables: condition
```

```
print(ANOVA_mem_score_test)
```

```
## Anova Table (Type 3 tests)
##
## Response: mem_score
##                     Effect    df   MSE    F   ges p.value
## 1                condition 1, 92 20.45 0.00 <.001    .992
## 2           predictiveness 1, 92  8.52 0.07 <.001    .792
## 3 condition:predictiveness 1, 92  8.52 0.35  .001    .553
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```r
bay_ANOVA_mem_score_test <- anovaBF(formula = mem_score ~ condition*predictiveness + pNum,
        data = data.frame(mem_score_test),
        whichRandom = "pNum")
print(bay_ANOVA_mem_score_test)
```

```
## Bayes factor analysis
## --------------
## [1] condition + pNum                                              : 0.2332035   ±1.05%
## [2] predictiveness + pNum                                         : 0.1674013   ±4.27%
## [3] condition + predictiveness + pNum                             : 0.03860993  ±3.21%
## [4] condition + predictiveness + condition:predictiveness + pNum : 0.009371259 ±1.81%
##
## Against denominator:
##   mem_score ~ pNum
## ---
## Bayes factor type: BFlinearModel, JZS
```

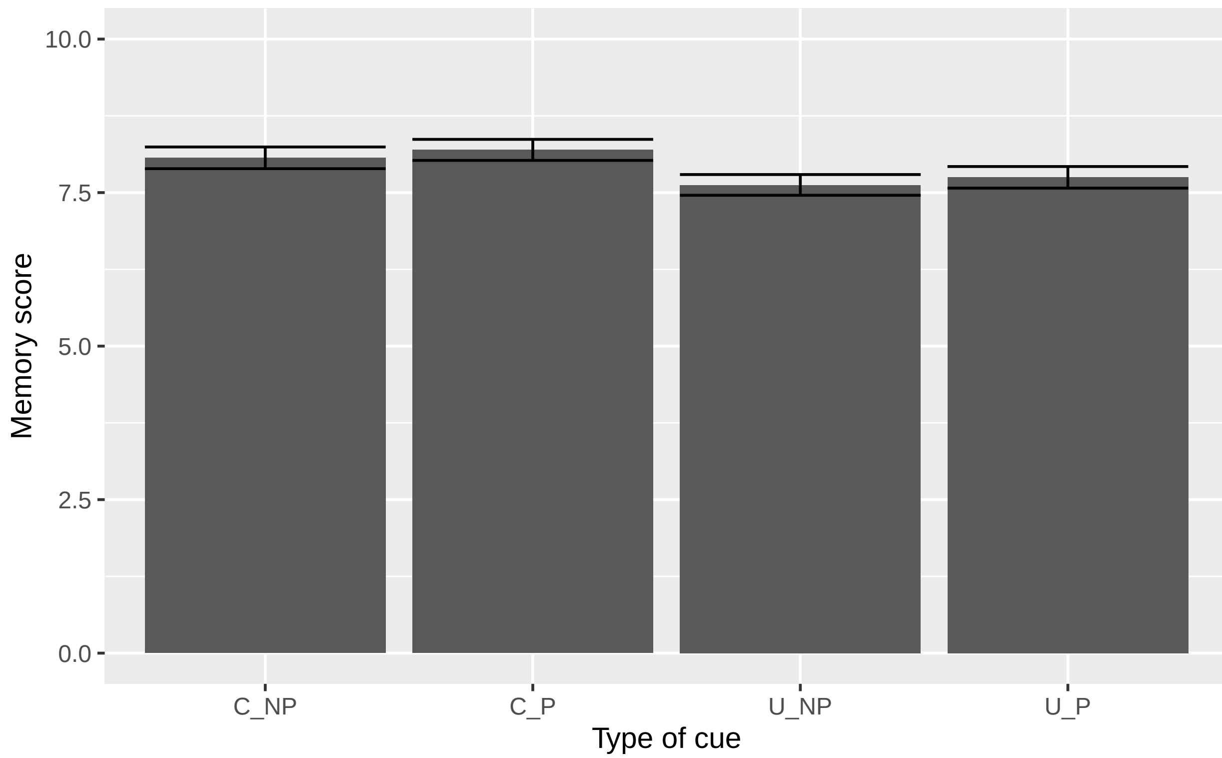
```r
bay_ANOVA_mem_score_test[4]/bay_ANOVA_mem_score_test[3]
```

```
## Bayes factor analysis
## --------------
## [1] condition + predictiveness + condition:predictiveness + pNum : 0.2427163 ±3.68%
##
## Against denominator:
##   mem_score ~ condition + predictiveness + pNum
## ---
## Bayes factor type: BFlinearModel, JZS
```

```r
#plot test mem_score but take out the errors
c_test <- filter(test, acc == 1)
c_m_mem_test <- c_test %>%
  group_by(cue_type) %>%
  summarise(mean_mem_score = mean(mem_score, na.rm = TRUE),
            sd_mem_score = sd(mem_score, na.rm = TRUE)/sqrt(length(mem_score)))
ggplot(data = c_m_mem_test) +
  geom_col(mapping = aes(x = cue_type, y = mean_mem_score)) +
  geom_errorbar(aes(x = cue_type, y= mean_mem_score, ymin = mean_mem_score - sd_mem_score, ymax = mean_
  coord_cartesian(ylim = c(0, 10))+
  scale_x_discrete (name = "Type of cue") +
  scale_y_continuous(name = "Memory score") +
  labs(title = "Figure 3", subtitle = "Mean memory score for each type of cue in test phase")
```

Figure 3

Mean memory score for each type of cue in test phase




```r
#ANOVA mem_score
c_mem_score_test <- c_test %>%
  group_by (pNum, condition, predictiveness) %>%
  summarise(mem_score = mean(mem_score, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'pNum', 'condition'. You can override using
## the `.groups` argument.
```

```r
c_mem_score_test$predictiveness <- factor(c_mem_score_test$predictiveness)
c_mem_score_test$condition <- factor(c_mem_score_test$condition)
c_mem_score_test$pNum <- factor(c_mem_score_test$pNum)
c_ANOVA_mem_score_test <- aov_car(formula = mem_score ~ condition + Error(pNum*predictiveness), data = c
```

```
## Warning: Missing values for 2 ID(s), which were removed before analysis:
## 79, 85
## Below the first few rows (in wide format) of the removed cases with missing data.
##      pNum condition non.predictive predictive
## # 79   79   certain             NA   4.872222
## # 85   85 uncertain             NA   5.578125
```

```
## Contrasts set to contr.sum for the following variables: condition
```

```r
print(c_ANOVA_mem_score_test)
```

```
## Anova Table (Type 3 tests)
##
## Response: mem_score
##                      Effect    df  MSE    F   ges p.value
```

```
## 1                condition 1, 90 4.78 2.36  .021    .128
## 2           predictiveness 1, 90 1.09 0.14 <.001    .708
## 3 condition:predictiveness 1, 90 1.09 0.12 <.001    .731
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
#interaction analysis
c_mem_score_interaction_p <- emmeans(c_ANOVA_mem_score_test, ~ predictiveness|condition)
pairs(c_mem_score_interaction_p, adjust = "bon")
```

```
## condition = certain:
##  contrast                     estimate    SE df t.ratio p.value
##  non.predictive - predictive -0.11124 0.223 90  -0.499  0.6188
##
## condition = uncertain:
##  contrast                     estimate    SE df t.ratio p.value
##  non.predictive - predictive -0.00478 0.213 90  -0.022  0.9822
```

```
c_mem_test_interaction_c <- emmeans(c_ANOVA_mem_score_test, ~ condition|predictiveness)
pairs(c_mem_test_interaction_c, adjust = "bon")
```

```
## predictiveness = non.predictive:
##  contrast            estimate    SE df t.ratio p.value
##  certain - uncertain    0.443 0.356 90   1.242  0.2174
##
## predictiveness = predictive:
##  contrast            estimate    SE df t.ratio p.value
##  certain - uncertain    0.549 0.359 90   1.529  0.1298
```

```
c_bay_ANOVA_mem_score_test <- anovaBF(formula = mem_score ~ condition*predictiveness + pNum,
        data = data.frame(c_mem_score_test),
        whichRandom = "pNum")
print(c_bay_ANOVA_mem_score_test)
```

```
## Bayes factor analysis
## --------------
## [1] condition + pNum                                              : 0.6926524  ±0.8%
## [2] predictiveness + pNum                                         : 0.1597472  ±0.91%
## [3] condition + predictiveness + pNum                             : 0.112457   ±2.07%
## [4] condition + predictiveness + condition:predictiveness + pNum : 0.02686986 ±4.08%
##
## Against denominator:
##   mem_score ~ pNum
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
c_bay_ANOVA_mem_score_test[4]/c_bay_ANOVA_mem_score_test[3]
```

```
## Bayes factor analysis
## --------------
## [1] condition + predictiveness + condition:predictiveness + pNum : 0.2389345 ±4.58%
##
## Against denominator:
##   mem_score ~ condition + predictiveness + pNum
## ---
## Bayes factor type: BFlinearModel, JZS
```