CSQ_Qualtrics

2022-03-23

```
#Calculate the CSQ score weightings according to Kennedy et al 1993
CSQConversion <- CSQdata


CSQConversion <- CSQConversion %>%
  mutate(NauseaSum =
           (CSQConversion$`General Discomfort` +
            CSQConversion$`Increased Salivation` +
            CSQConversion$Sweating +
            CSQConversion$Nausea +
            CSQConversion$`Difficulty Concentrating`+
            CSQConversion$`Stomach Awareness` +
            CSQConversion$Burping)) %>%
  mutate(OculomotorSum =
           (CSQConversion$`General Discomfort` +
            CSQConversion$Fatigue +
            CSQConversion$Headache +
            CSQConversion$`Eye Strain` +
            CSQConversion$`Difficulty Focusing` +
            CSQConversion$`Difficulty Concentrating`+
            CSQConversion$`Blurred Vision`)) %>%
  mutate(DisorientationSum =
           (CSQConversion$`Difficulty Focusing` +
            CSQConversion$Nausea +
            CSQConversion$`Fullness of Head` +
            CSQConversion$`Blurred Vision` +
            CSQConversion$`Dizzy (Eyes Open)` +
            CSQConversion$`Dizzy ( Eyes Closed)`+
            CSQConversion$Vertigo))

#Calculate the final weighted CSQ scores

  CSQConversion <- CSQConversion %>%
      mutate(NauseaWeight = (CSQConversion$NauseaSum * 9.54)) %>%
      mutate(OculomotorWeight = (CSQConversion$OculomotorSum*7.58)) %>%
      mutate(DisorientationWeight = (CSQConversion$DisorientationSum *13.92)) %>%
      mutate(TotalCSQScore =
                (CSQConversion$NauseaSum +
                CSQConversion$OculomotorSum +
                CSQConversion$DisorientationSum)*
                3.74)
```

```
#Summarise the descriptives for the converted scores
CSQConversionSummary <- CSQConversion %>%
```

```
  group_by(Condition, Timepoint) %>%
  summarise(
  n = n(),
  mean = mean(TotalCSQScore),
  sd = sd(TotalCSQScore),
  min = min(TotalCSQScore),
  max = max(TotalCSQScore)

  )
```

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

```
CSQConversionSummary
```

```
## # A tibble: 6 x 7
## # Groups:   Condition [2]
##   Condition Timepoint     n  mean    sd   min   max
##   <chr>     <chr>     <int> <dbl> <dbl> <dbl> <dbl>
## 1 Global    1            21 16.9  25.7      0 101.
## 2 Global    2            21 13.0  14.9      0  52.4
## 3 Global    3            21 13.7  21.1      0  78.5
## 4 Local     1            21  7.12  9.15     0  37.4
## 5 Local     2            21  9.08 10.4      0  33.7
## 6 Local     3            21  9.80  8.89     0  33.7
```

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

```
## # A tibble: 6 x 7
## # Groups:   Condition [2]
##   Condition Timepoint     n  mean    sd   min   max
##   <chr>     <chr>     <int> <dbl> <dbl> <dbl> <dbl>
## 1 Global    1            21 13.6  21.5      0  85.9
## 2 Global    2            21  6.81 10.1      0  38.2
## 3 Global    3            21  7.72 15.0      0  57.2
## 4 Local     1            21  5.91 12.2      0  47.7
## 5 Local     2            21  4.09  7.73     0  28.6
## 6 Local     3            21  2.73  5.35     0  19.1
```

```
#Summarise the descriptives for the converted Oculomotor scores
CSQOculoSummary <- CSQConversion %>%

  group_by(Condition, Timepoint) %>%
  summarise(
  n = n(),
  mean = mean(OculomotorWeight),
  sd = sd(OculomotorWeight),
  min = min(OculomotorWeight),
  max = max(OculomotorWeight)

  )
```

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

```
CSQOculoSummary
```

```
## # A tibble: 6 x 7
## # Groups:   Condition [2]
##   Condition Timepoint     n  mean    sd   min   max
##   <chr>     <chr>     <int> <dbl> <dbl> <dbl> <dbl>
## 1 Global    1            21 15.9  22.9      0  83.4
## 2 Global    2            21 15.2  15.7      0  60.6
## 3 Global    3            21 16.6  22.4      0  91.0
## 4 Local     1            21  9.02  9.77     0  30.3
## 5 Local     2            21 12.3  13.0      0  45.5
## 6 Local     3            21 13.0  11.3      0  37.9
```

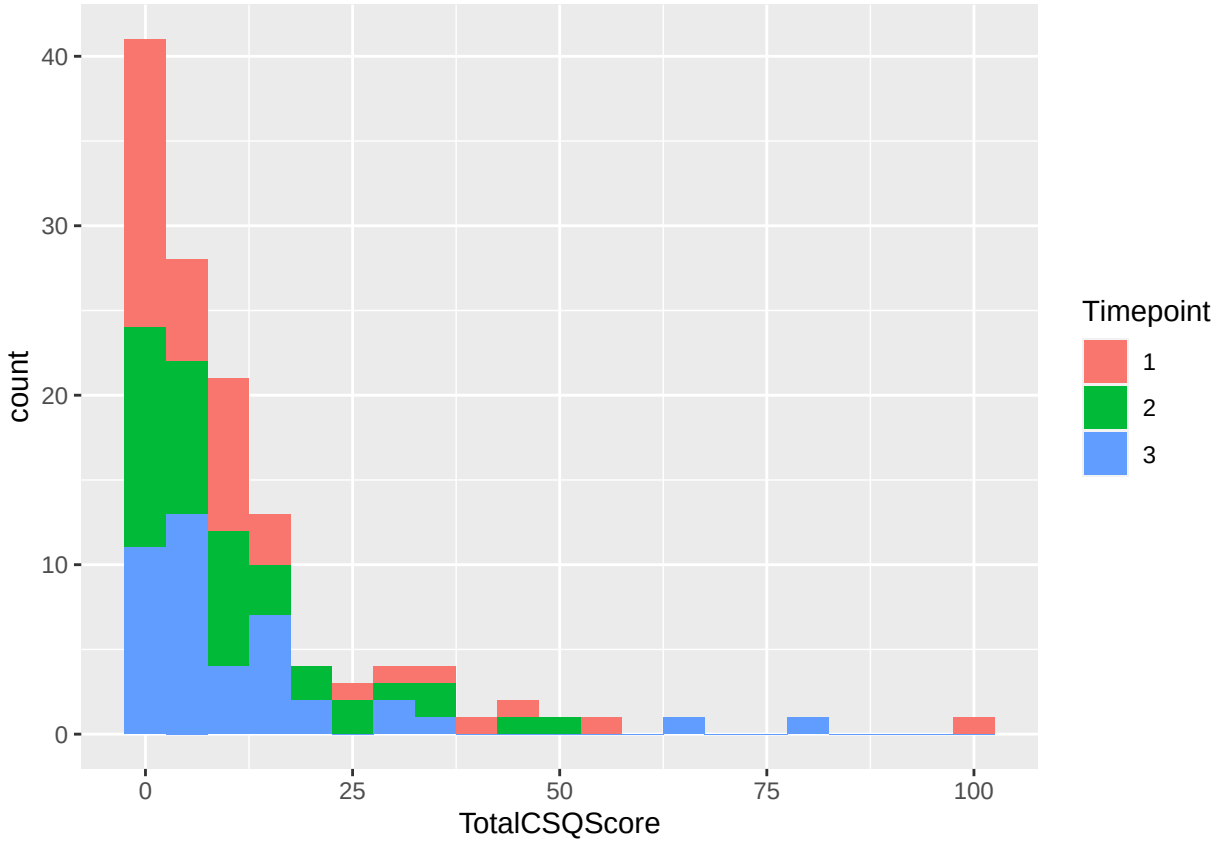
```
#Summarise the descriptives for the converted Disorientation scores
CSQDisoriSummary <- CSQConversion %>%

  group_by(Condition, Timepoint) %>%
  summarise(
  n = n(),
  mean = mean(DisorientationWeight),
  sd = sd(DisorientationWeight),
  min = min(DisorientationWeight),
  max = max(DisorientationWeight)

  )
```

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

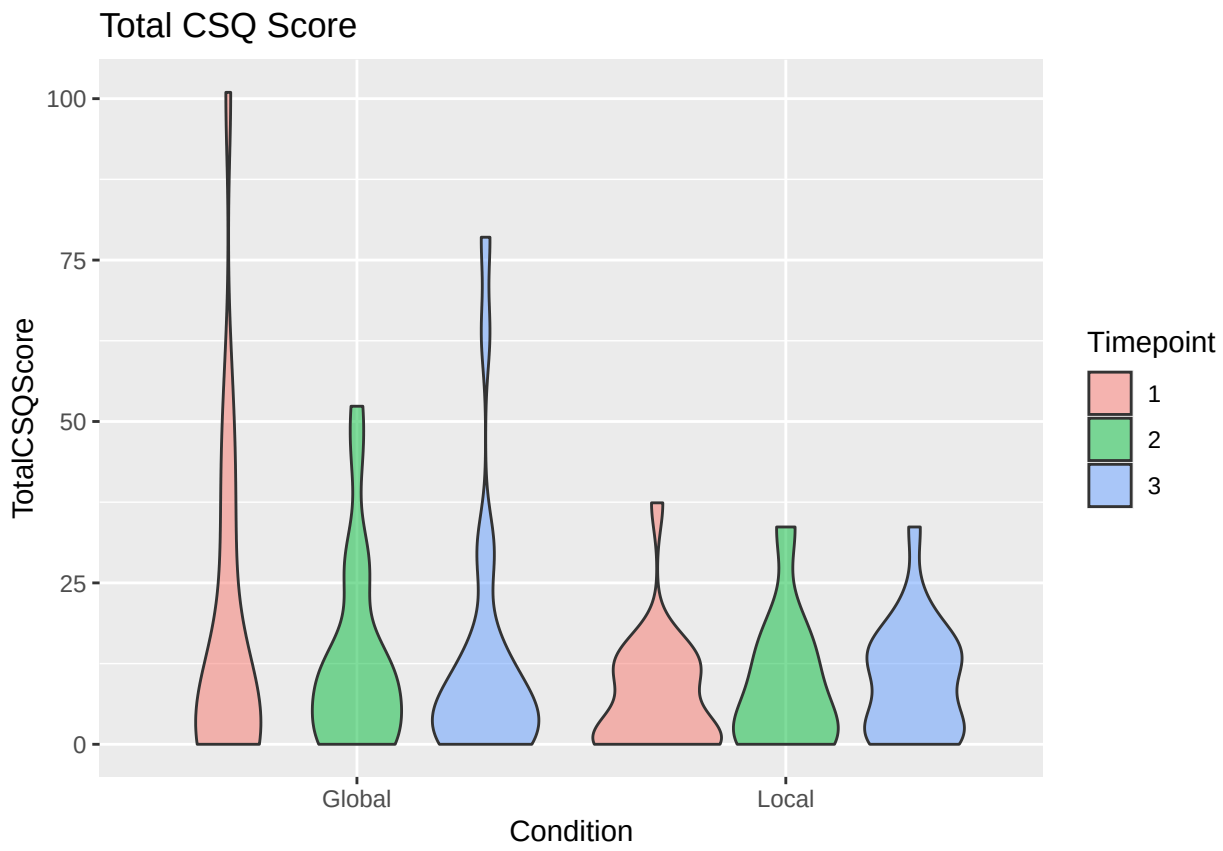
```
CSQDisoriSummary
```

```
## # A tibble: 6 x 7
## # Groups:   Condition [2]
##   Condition Timepoint     n  mean    sd   min   max
##   <chr>     <chr>     <int> <dbl> <dbl> <dbl> <dbl>
## 1 Global    1            21 13.9  26.0      0  97.4
## 2 Global    2            21 10.6  18.1      0  69.6
## 3 Global    3            21  9.28 19.4      0  69.6
## 4 Local     1            21  1.33  4.19     0  13.9
## 5 Local     2            21  5.30  9.31     0  27.8
## 6 Local     3            21  8.62 11.2      0  27.8
```

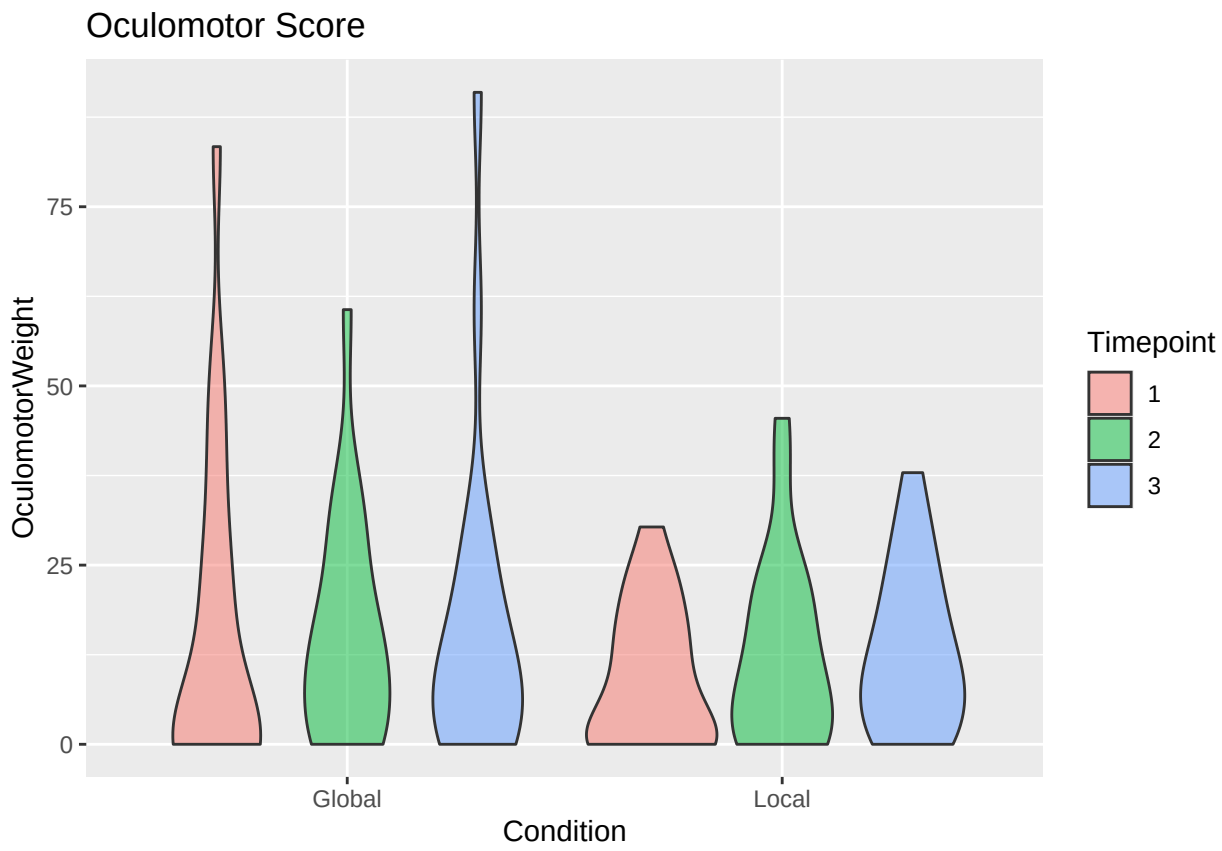
```
#Visualise the spread of the data
ggplot(CSQConversion, aes(x = TotalCSQScore, fill = Timepoint ))+
  geom_histogram(binwidth = 5)
```



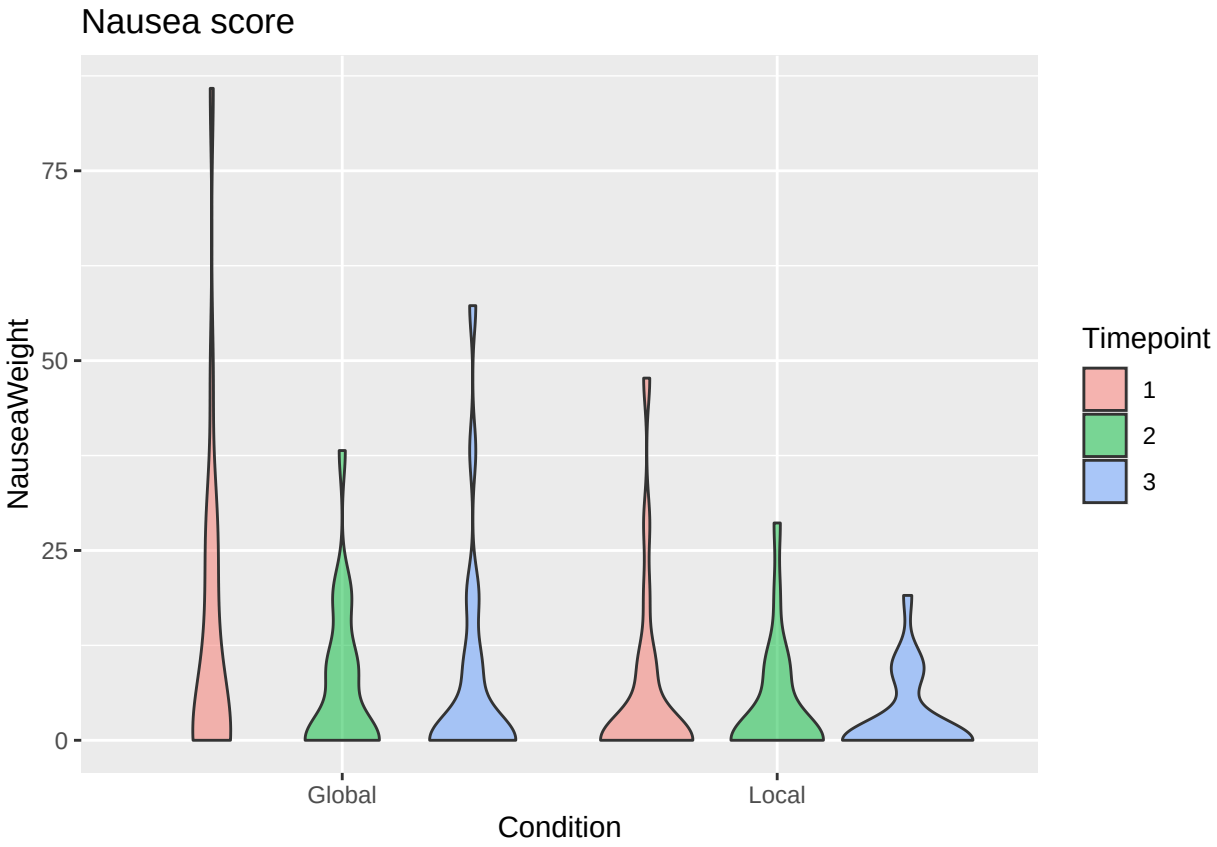

```
#Create violin plots for each of the Weighted totals to show the spread of the data
ggplot(CSQConversion, aes(Condition, TotalCSQScore, fill = Timepoint ))+
  geom_violin(alpha = 0.5)+
  labs(title = "Total CSQ Score")
```



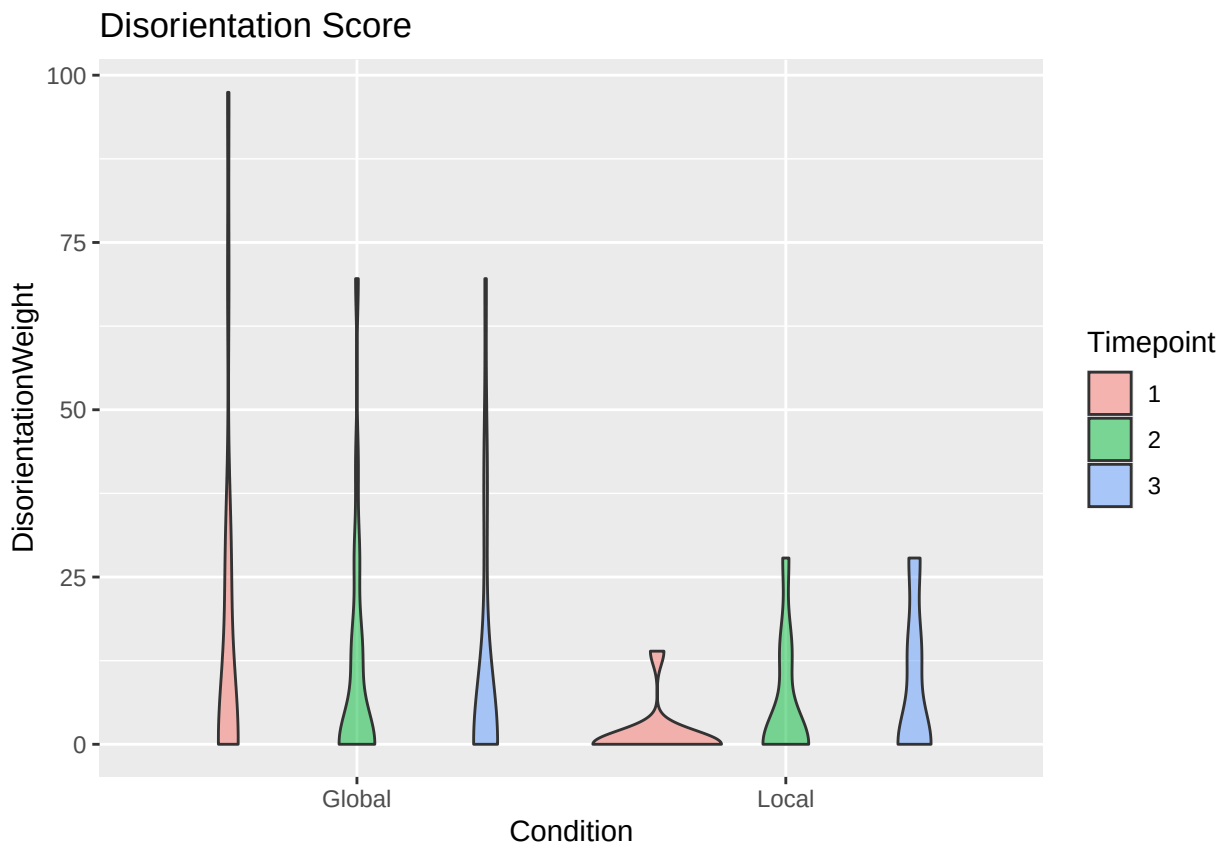

```
ggplot(CSQConversion, aes(Condition, OculomotorWeight, fill = Timepoint ))+
  geom_violin(alpha = 0.5) +
  labs(title = "Oculomotor Score")
```



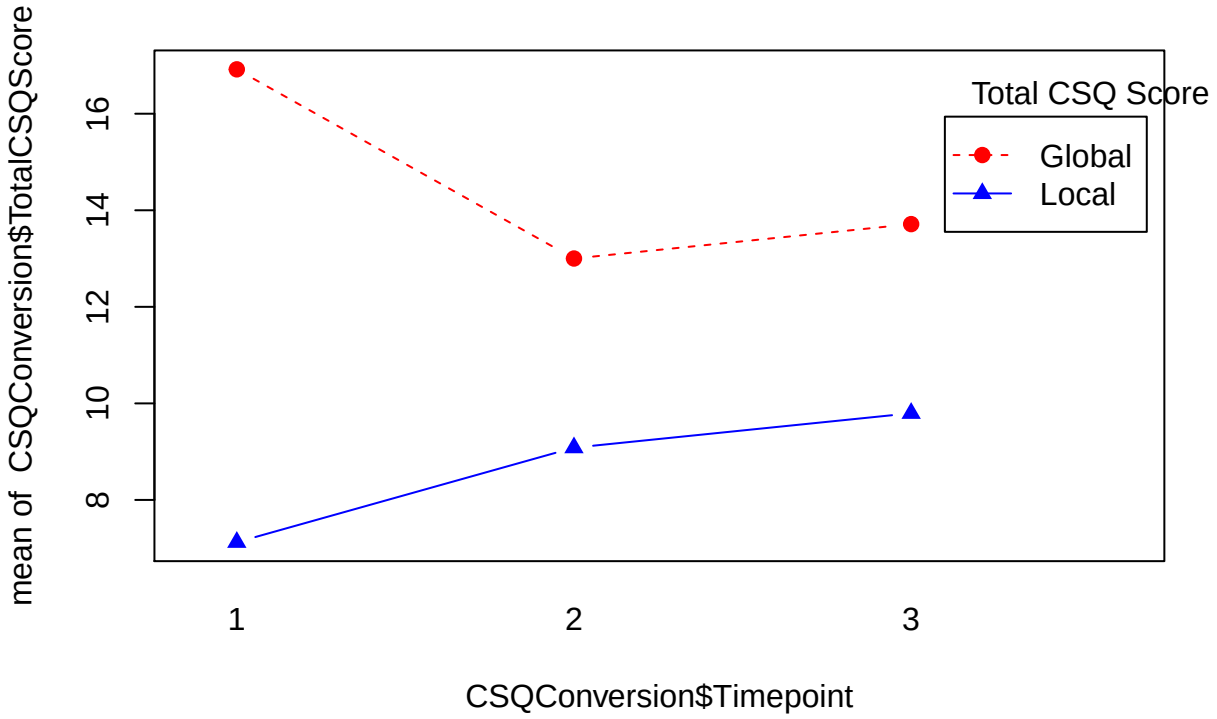

```
ggplot(CSQConversion, aes(Condition, NauseaWeight, fill = Timepoint ))+
  geom_violin(alpha = 0.5) +
  labs(title = "Nausea score")
```



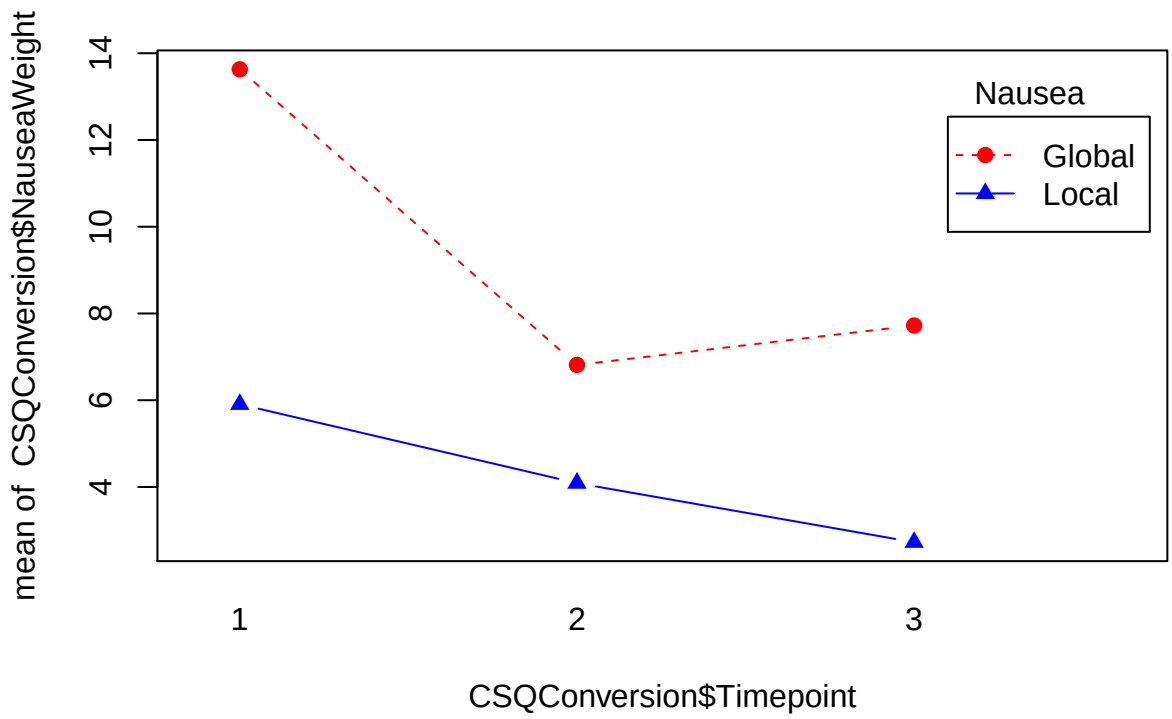

```
ggplot(CSQConversion, aes(Condition, DisorientationWeight, fill = Timepoint ))+
  geom_violin(alpha = 0.5)+
  labs(title = "Disorientation Score")
```



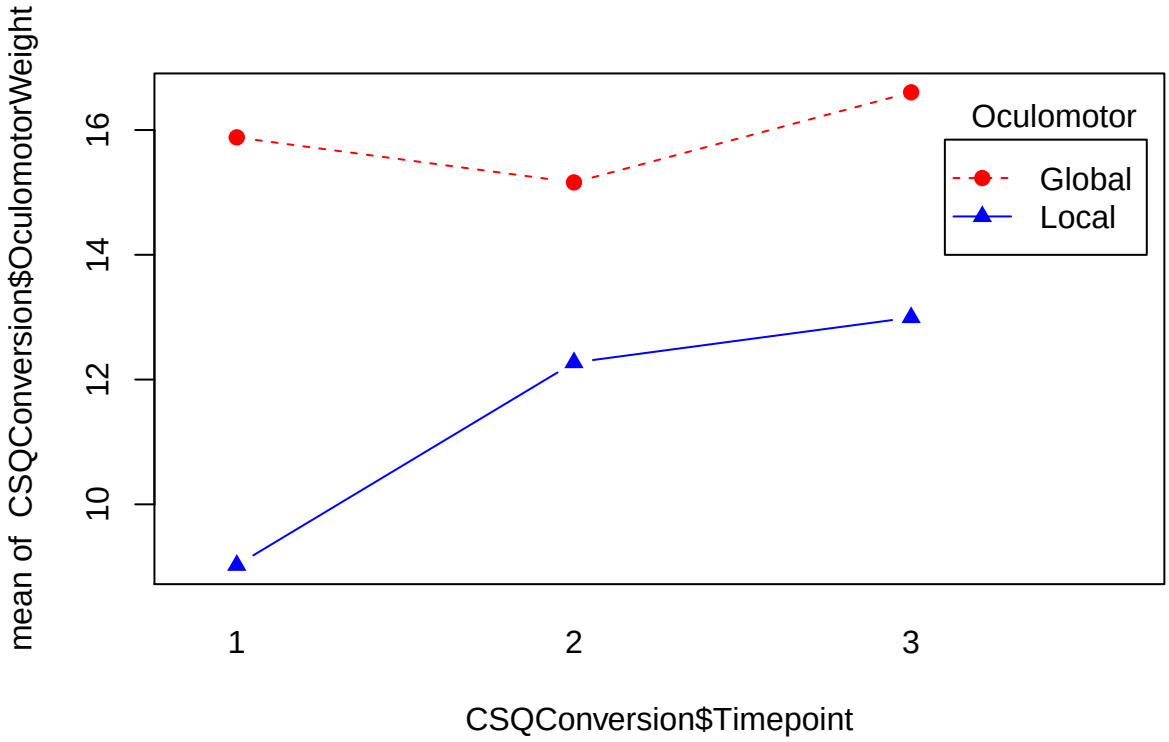

```
#Create interaction plots for the mean of each condition at each of the 3 timepoints
#**pch = type of shape on the point graph
interaction.plot(x.factor =  CSQConversion$Timepoint,
                 trace.factor = CSQConversion$Condition,
                 response = CSQConversion$TotalCSQScore,
                 fun = mean,
                 type= "b",
                 col=c("red","blue"),
                 pch=c(19,17,15),
                 trace.label = "Total CSQ Score",
                 fixed=T,
                 leg.bty ="o")
```



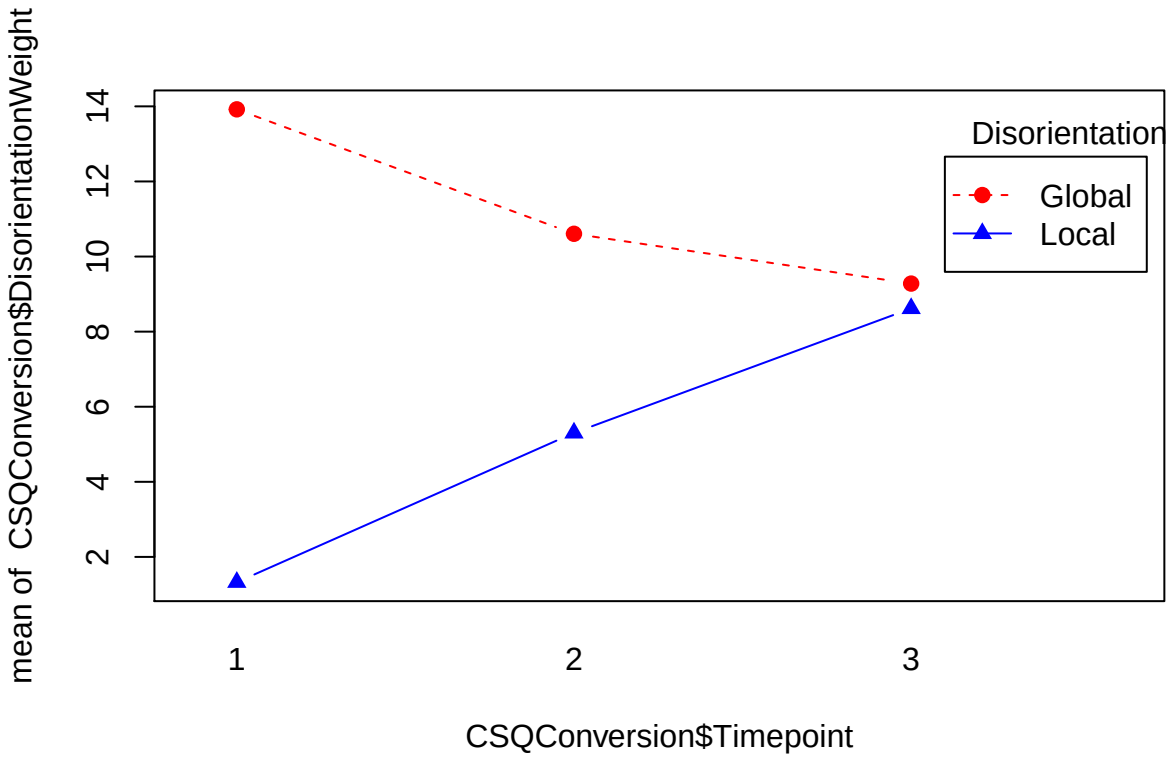

```
interaction.plot(x.factor =  CSQConversion$Timepoint,
                 trace.factor = CSQConversion$Condition,
                 response = CSQConversion$NauseaWeight,
                 fun = mean,
                 type= "b",
                 col=c("red","blue"),
                 pch=c(19,17,15),
                 trace.label = "Nausea",
                 fixed=T,
                 leg.bty ="o")
```




```
interaction.plot(x.factor =  CSQConversion$Timepoint,
                 trace.factor = CSQConversion$Condition,
                 response = CSQConversion$OculomotorWeight,
                 fun = mean,
                 type= "b",
                 col=c("red","blue"),
                 pch=c(19,17,15),
                 trace.label = "Oculomotor",
                 fixed=T,
                 leg.bty ="o")
```




```
interaction.plot(x.factor =  CSQConversion$Timepoint,
                 trace.factor = CSQConversion$Condition,
                 response = CSQConversion$DisorientationWeight,
                 fun = mean,
                 type= "b",
                 col=c("red","blue"),
                 pch=c(19,17,15),
                 trace.label = "Disorientation",
                 fixed=T,
                 leg.bty ="o")
```




```
# Generate linear modals for each of the weighted scores
CSQModel = lm(TotalCSQScore ~ Timepoint + Condition + Timepoint:Condition,
              data = CSQConversion)

Anova(CSQModel,
      type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: TotalCSQScore
##                      Sum Sq  Df F value  Pr(>F)
## Timepoint               22   2  0.0404 0.96039
## Condition             1088   1  4.0849 0.04549 *
## Timepoint:Condition    242   2  0.4539 0.63624
## Residuals            31962 120
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Calculate comparison means for Overall scores
CSQmarginal = emmeans(CSQModel, ~Condition:Timepoint)
pairs(CSQmarginal, adjust="tukey")
```

```
##  contrast            estimate   SE  df t.ratio p.value
##  Global 1 - Local 1     9.795 5.04 120   1.945  0.3804
##  Global 1 - Global 2    3.918 5.04 120   0.778  0.9708
```

```
##  Global 1 - Local 2     7.836 5.04 120   1.556  0.6290
##  Global 1 - Global 3    3.206 5.04 120   0.636  0.9880
##  Global 1 - Local 3     7.124 5.04 120   1.414  0.7183
##  Local 1 - Global 2    -5.877 5.04 120  -1.167  0.8516
##  Local 1 - Local 2     -1.959 5.04 120  -0.389  0.9988
##  Local 1 - Global 3    -6.590 5.04 120  -1.308  0.7799
##  Local 1 - Local 3     -2.671 5.04 120  -0.530  0.9948
##  Global 2 - Local 2     3.918 5.04 120   0.778  0.9708
##  Global 2 - Global 3   -0.712 5.04 120  -0.141  1.0000
##  Global 2 - Local 3     3.206 5.04 120   0.636  0.9880
##  Local 2 - Global 3    -4.630 5.04 120  -0.919  0.9408
##  Local 2 - Local 3     -0.712 5.04 120  -0.141  1.0000
##  Global 3 - Local 3     3.918 5.04 120   0.778  0.9708
##
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
# Generate linear modals for Oculomotor subscale

OcuModel = lm(OculomotorWeight ~ Timepoint + Condition + Timepoint:Condition,
              data = CSQConversion)

Anova(OcuModel,
      type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: OculomotorWeight
##                      Sum Sq  Df F value Pr(>F)
## Timepoint               116   2  0.2087 0.8119
## Condition               624   1  2.2498 0.1363
## Timepoint:Condition      94   2  0.1693 0.8445
## Residuals             33297 120
```

```
#Calculate comparison means for Oculomotor subscale

Ocumarginal = emmeans(OcuModel, ~Condition:Timepoint)
pairs(Ocumarginal, adjust="tukey")
```

```
##  contrast            estimate   SE  df t.ratio p.value
##  Global 1 - Local 1     6.858 5.14 120   1.334  0.7655
##  Global 1 - Global 2    0.722 5.14 120   0.140  1.0000
##  Global 1 - Local 2     3.610 5.14 120   0.702  0.9814
##  Global 1 - Global 3   -0.722 5.14 120  -0.140  1.0000
##  Global 1 - Local 3     2.888 5.14 120   0.562  0.9933
##  Local 1 - Global 2    -6.136 5.14 120  -1.194  0.8391
##  Local 1 - Local 2     -3.249 5.14 120  -0.632  0.9884
##  Local 1 - Global 3    -7.580 5.14 120  -1.475  0.6811
##  Local 1 - Local 3     -3.970 5.14 120  -0.772  0.9717
##  Global 2 - Local 2     2.888 5.14 120   0.562  0.9933
##  Global 2 - Global 3   -1.444 5.14 120  -0.281  0.9998
##  Global 2 - Local 3     2.166 5.14 120   0.421  0.9983
##  Local 2 - Global 3    -4.331 5.14 120  -0.843  0.9588
##  Local 2 - Local 3     -0.722 5.14 120  -0.140  1.0000
```

```
##  Global 3 - Local 3     3.610 5.14 120   0.702  0.9814
##
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
# Generate linear modals for Nausea subscale
NauModel = lm(NauseaWeight ~ Timepoint + Condition + Timepoint:Condition,
              data = CSQConversion)

Anova(NauModel,
      type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: NauseaWeight
##                       Sum Sq  Df F value  Pr(>F)
## Timepoint              550.4   2  1.6124 0.20371
## Condition              835.0   1  4.8921 0.02887 *
## Timepoint:Condition    131.5   2  0.3851 0.68122
## Residuals            20481.9 120
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Calculate comparison means for Nausea subscale

Naumarginal = emmeans(NauModel, ~Condition:Timepoint)
pairs(Naumarginal, adjust="tukey")
```

```
##  contrast            estimate   SE  df t.ratio p.value
##  Global 1 - Local 1     7.723 4.03 120   1.915  0.3979
##  Global 1 - Global 2    6.814 4.03 120   1.690  0.5410
##  Global 1 - Local 2     9.540 4.03 120   2.366  0.1767
##  Global 1 - Global 3    5.906 4.03 120   1.465  0.6872
##  Global 1 - Local 3    10.903 4.03 120   2.704  0.0819
##  Local 1 - Global 2    -0.909 4.03 120  -0.225  0.9999
##  Local 1 - Local 2      1.817 4.03 120   0.451  0.9976
##  Local 1 - Global 3    -1.817 4.03 120  -0.451  0.9976
##  Local 1 - Local 3      3.180 4.03 120   0.789  0.9690
##  Global 2 - Local 2     2.726 4.03 120   0.676  0.9843
##  Global 2 - Global 3   -0.909 4.03 120  -0.225  0.9999
##  Global 2 - Local 3     4.089 4.03 120   1.014  0.9124
##  Local 2 - Global 3    -3.634 4.03 120  -0.901  0.9455
##  Local 2 - Local 3      1.363 4.03 120   0.338  0.9994
##  Global 3 - Local 3     4.997 4.03 120   1.239  0.8166
##
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
#Significatn effect of conidtion, but is meaningless in the comparisons
```

```
# Generate linear modals for Disorientation  subscale
DisModel = lm(DisorientationWeight ~ Timepoint + Condition + Timepoint:Condition,
              data = CSQConversion)
```

```
Anova(DisModel,
      type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: DisorientationWeight
##                      Sum Sq  Df F value Pr(>F)
## Timepoint               40   2  0.0745 0.9283
## Condition             1206   1  4.4928 0.0361 *
## Timepoint:Condition    760   2  1.4155 0.2468
## Residuals            32202 120
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Calculate comparison means for Disorientation subscale

Dismarginal = emmeans(DisModel, ~Condition:Timepoint)
pairs(Dismarginal, adjust="tukey")
```

```
##  contrast            estimate   SE  df t.ratio p.value
##  Global 1 - Local 1    12.594 5.06 120   2.491  0.1349
##  Global 1 - Global 2    3.314 5.06 120   0.656  0.9863
##  Global 1 - Local 2     8.617 5.06 120   1.705  0.5315
##  Global 1 - Global 3    4.640 5.06 120   0.918  0.9413
##  Global 1 - Local 3     5.303 5.06 120   1.049  0.9002
##  Local 1 - Global 2    -9.280 5.06 120  -1.836  0.4471
##  Local 1 - Local 2     -3.977 5.06 120  -0.787  0.9693
##  Local 1 - Global 3    -7.954 5.06 120  -1.573  0.6176
##  Local 1 - Local 3     -7.291 5.06 120  -1.442  0.7012
##  Global 2 - Local 2     5.303 5.06 120   1.049  0.9002
##  Global 2 - Global 3    1.326 5.06 120   0.262  0.9998
##  Global 2 - Local 3     1.989 5.06 120   0.393  0.9988
##  Local 2 - Global 3    -3.977 5.06 120  -0.787  0.9693
##  Local 2 - Local 3     -3.314 5.06 120  -0.656  0.9863
##  Global 3 - Local 3     0.663 5.06 120   0.131  1.0000
##
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
# Calculate quartiles for outlier removal
quantile(CSQConversion$TotalCSQScore, probs=c(.25,.50,.75, .90, .98), na.rm = FALSE)
```

```
##   25%   50%   75%   90%   98%
##  0.00  7.48 14.96 29.92 59.84
```

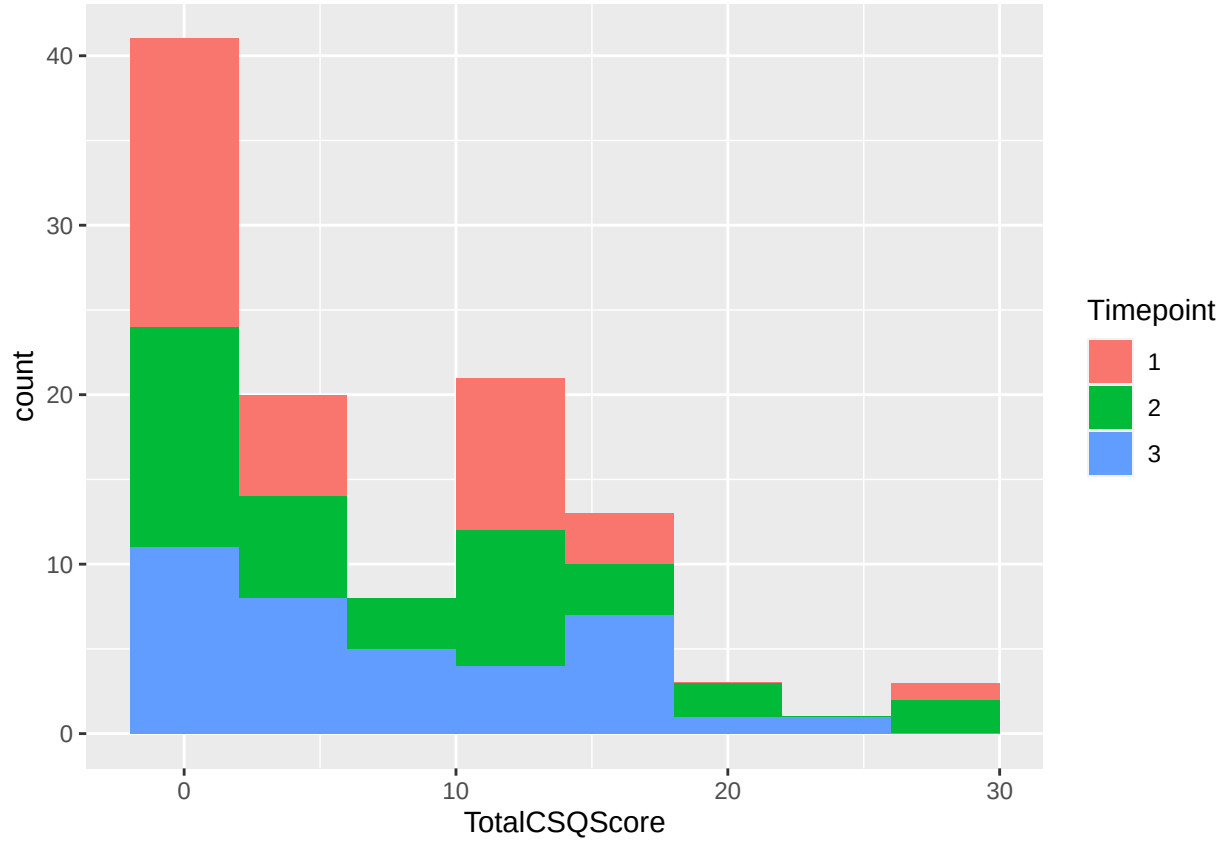
```
IQR(CSQConversion$TotalCSQScore)
```

```
## [1] 14.96
```

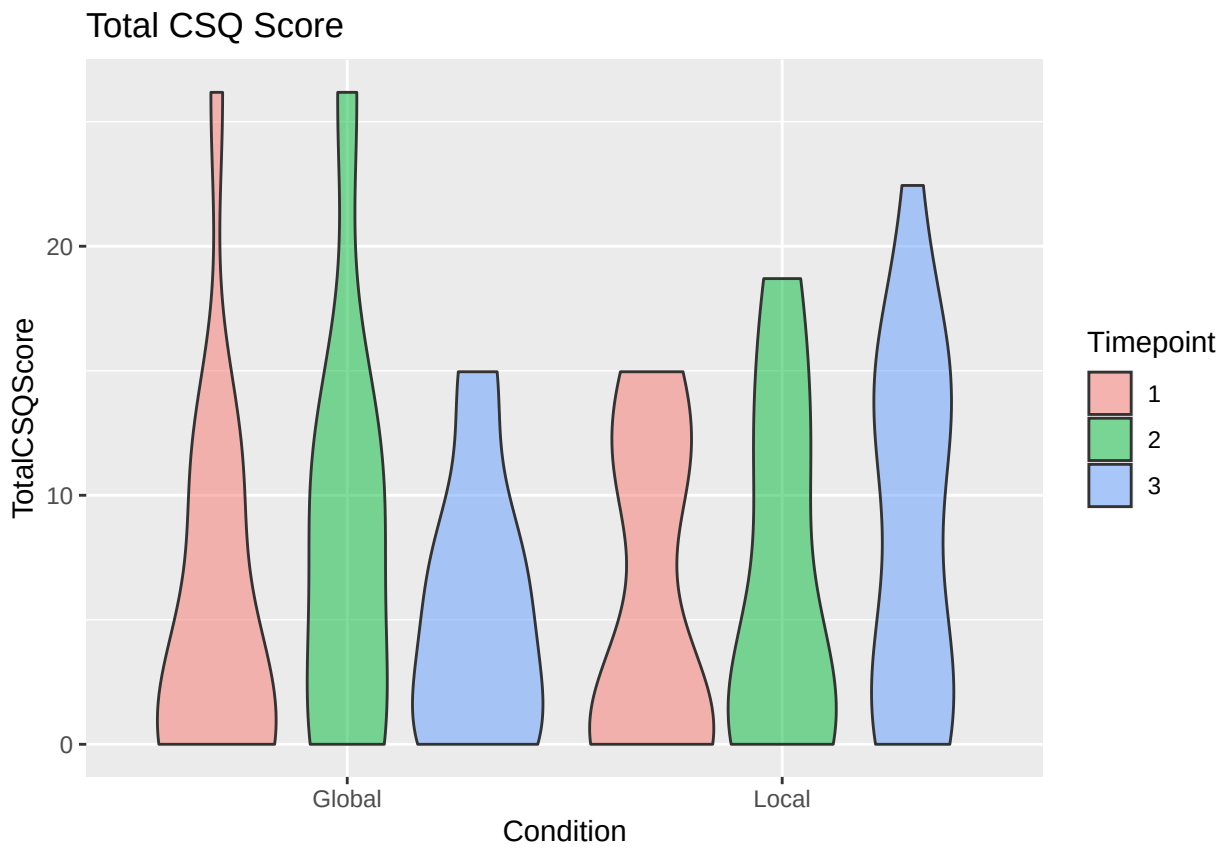
```
CSQConversionOutliersRM <- CSQConversion %>%
#Remove outlier participants
```

```
subset(TotalCSQScore < quantile(CSQConversion$TotalCSQScore, probs = .90), na.rm = F)


# subset(ID != "P07") %>%
# subset(ID != "P13") %>%
# subset(ID != "P16") %>%
# subset(ID != "P19") %>%
# subset(ID != "P21")

# CSQDataZScores <- CSQdata %>%
#   group_by(Condition, Timepoint)


#z_scores <- as.data.frame(sapply(df, function(df) (abs(df-mean(df))/sd(df))))
```

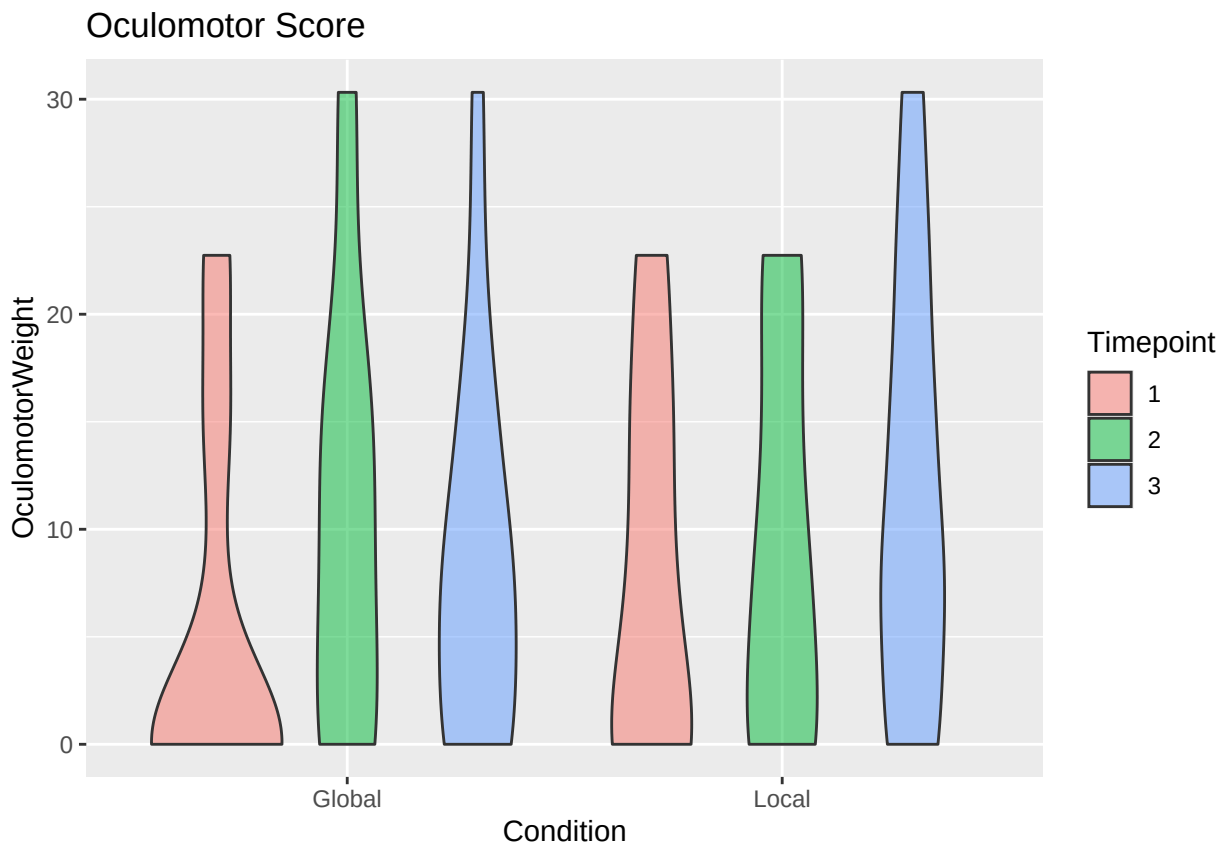
```
#SPread of data with outliers >90% removed
ggplot(CSQConversionOutliersRM, aes(x = TotalCSQScore, fill = Timepoint ))+
  geom_histogram(binwidth = 4)
```



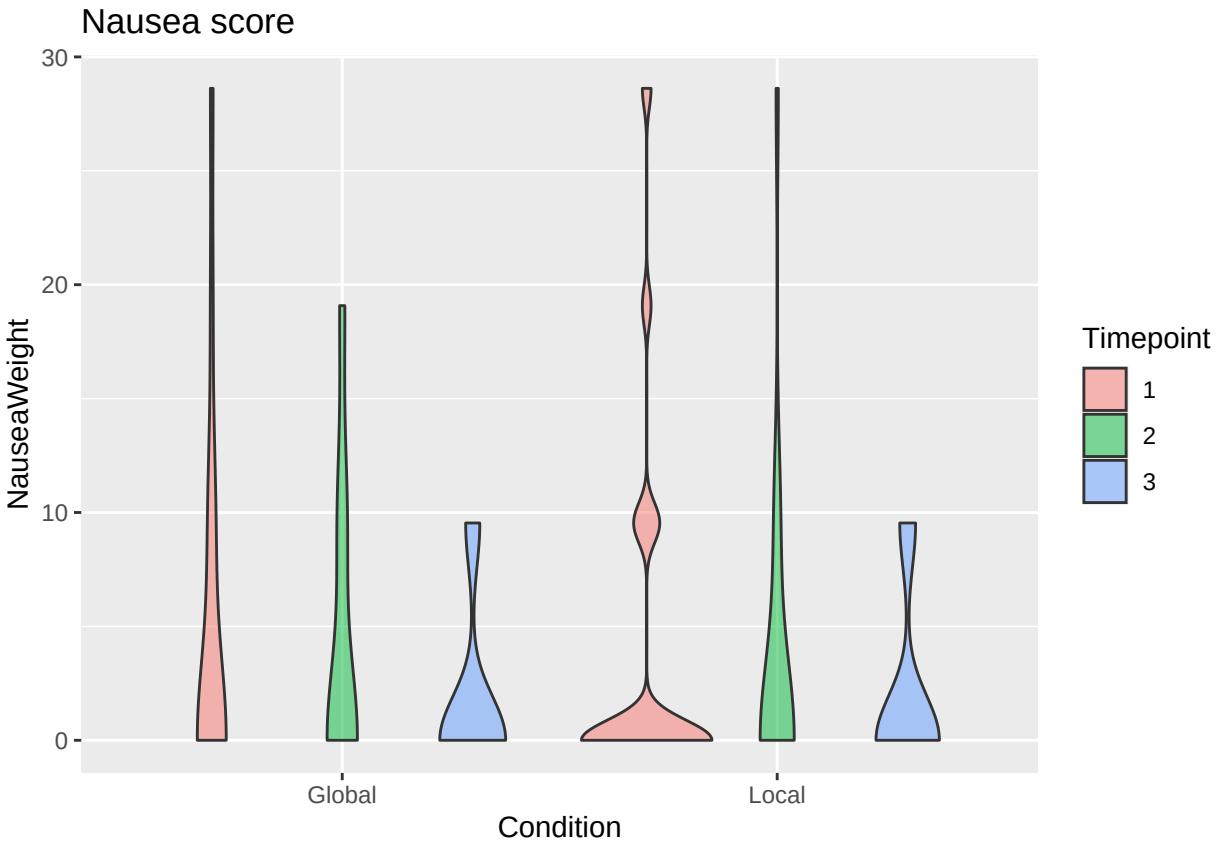

```
#Violing plots for outliers >90% removed

ggplot(CSQConversionOutliersRM, aes(Condition, TotalCSQScore, fill = Timepoint ))+
  geom_violin(alpha = 0.5)+
  labs(title = "Total CSQ Score")
```



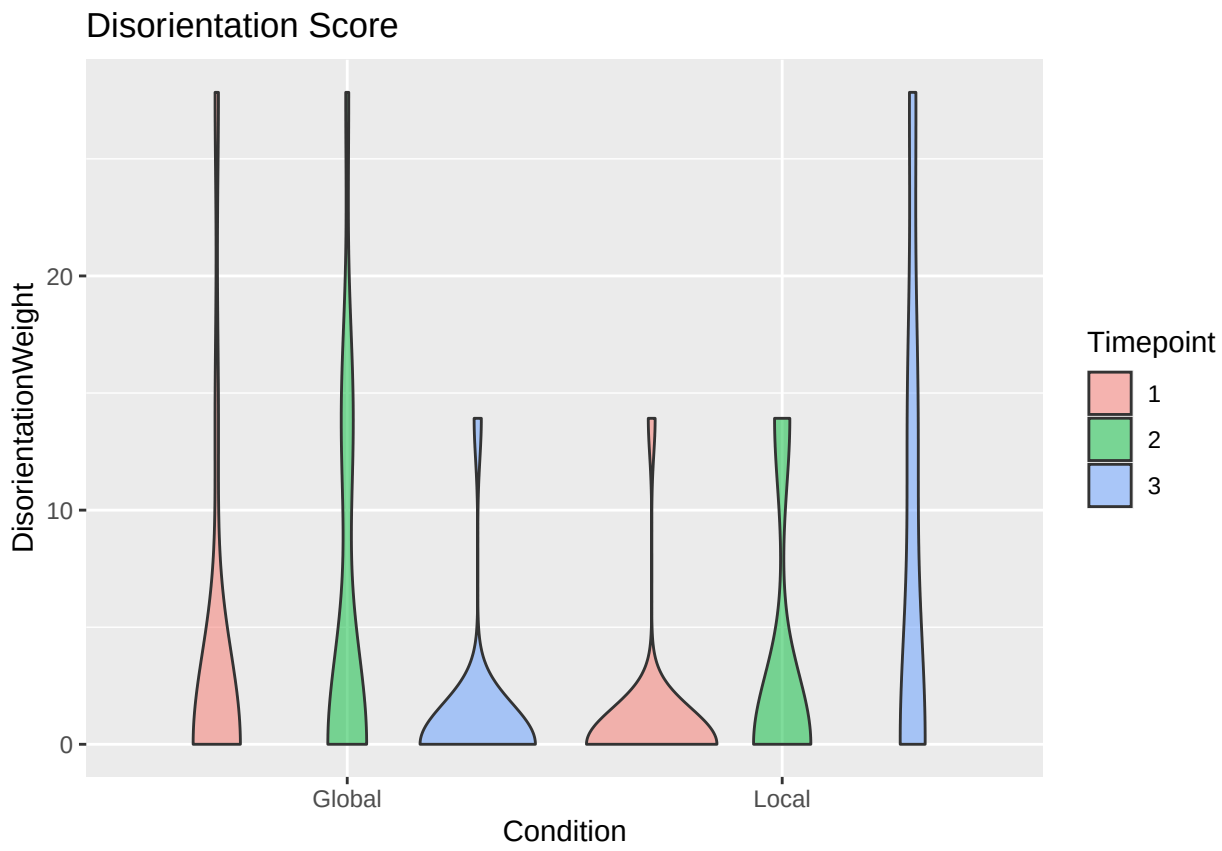

```
ggplot(CSQConversionOutliersRM, aes(Condition, OculomotorWeight, fill = Timepoint ))+
  geom_violin(alpha = 0.5) +
  labs(title = "Oculomotor Score")
```



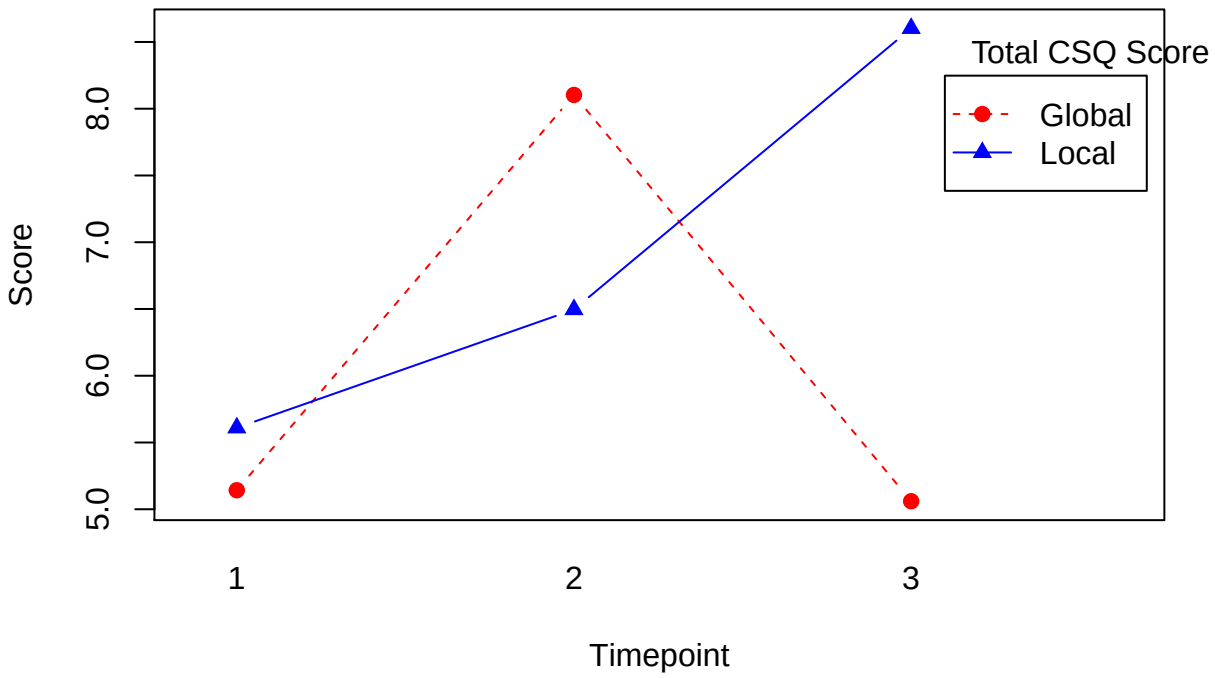

```
ggplot(CSQConversionOutliersRM, aes(Condition, NauseaWeight, fill = Timepoint ))+
  geom_violin(alpha = 0.5) +
  labs(title = "Nausea score")
```



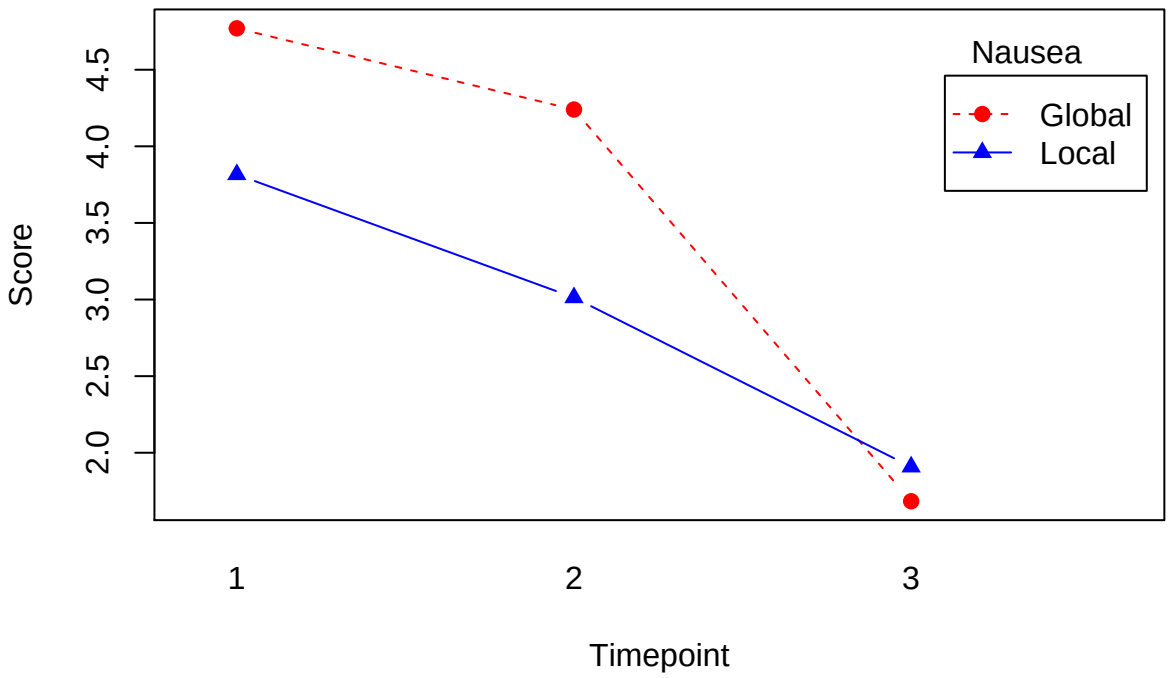

```
ggplot(CSQConversionOutliersRM, aes(Condition, DisorientationWeight, fill = Timepoint ))+
  geom_violin(alpha = 0.5)+
  labs(title = "Disorientation Score")
```



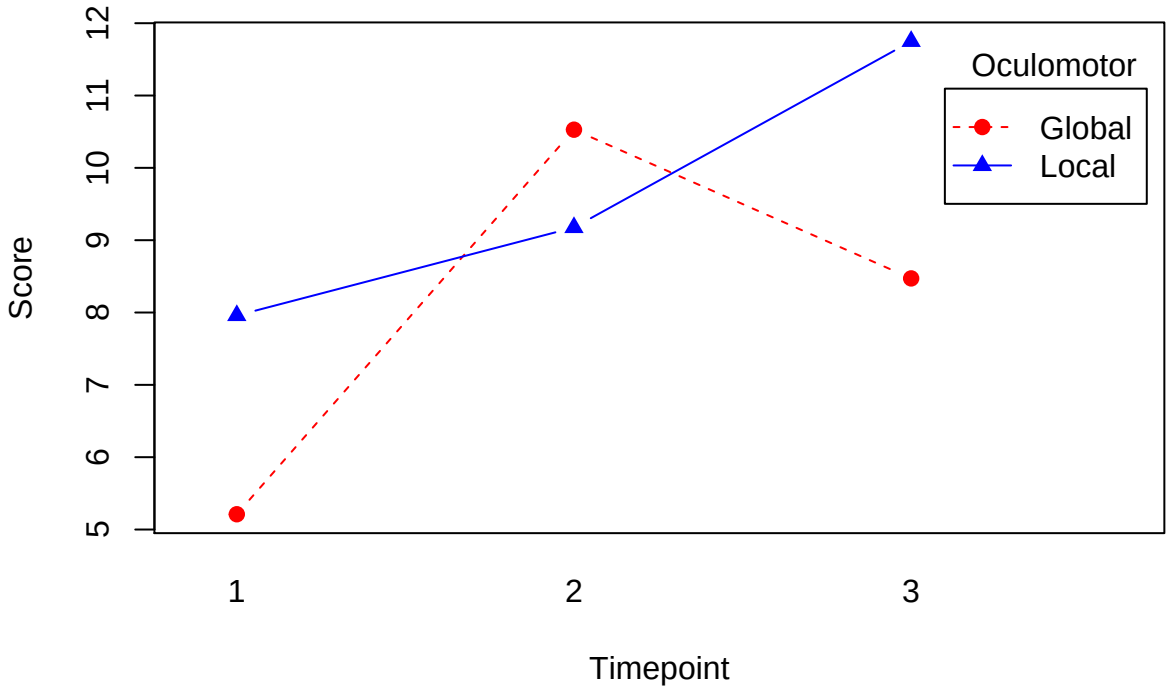

```
#interaction plots for >90% removed
interaction.plot(x.factor =  CSQConversionOutliersRM$Timepoint,
                 trace.factor = CSQConversionOutliersRM$Condition,
                 response = CSQConversionOutliersRM$TotalCSQScore,
                 fun = mean,
                 xlab = "Timepoint",
                 ylab = "Score",
                 type= "b",
                 col=c("red","blue"),
                 pch=c(19,17,15),
                 trace.label = "Total CSQ Score",
                 fixed=T,
                 leg.bty ="o")
```



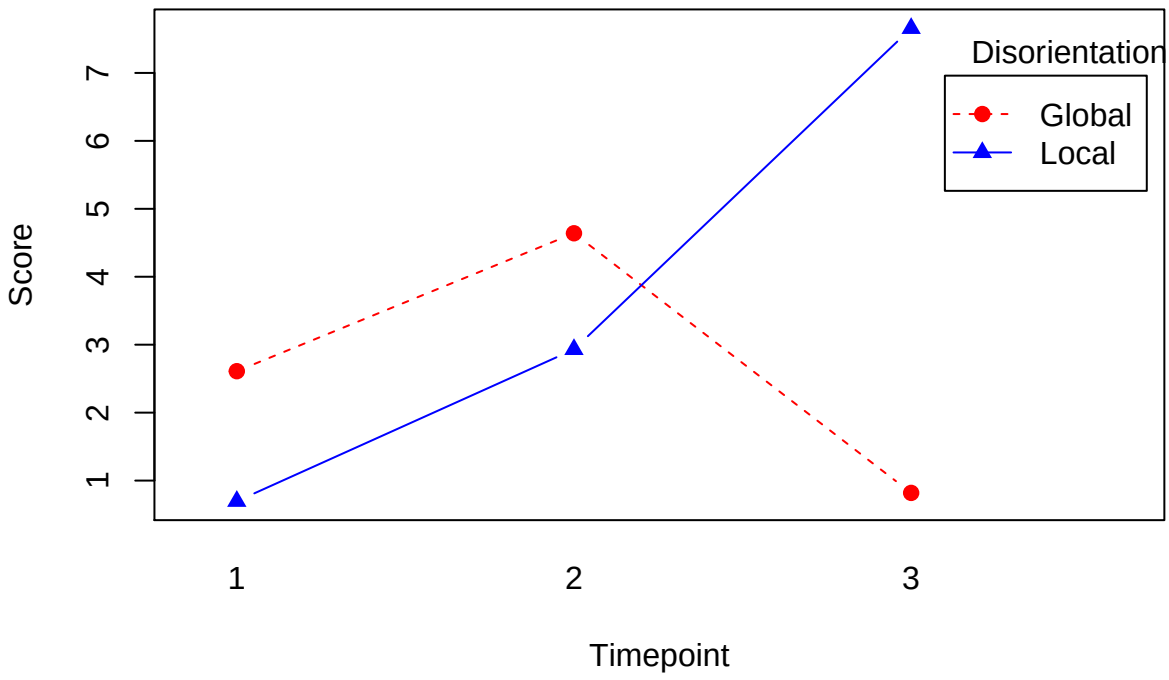

```
interaction.plot(x.factor =  CSQConversionOutliersRM$Timepoint,
                 trace.factor = CSQConversionOutliersRM$Condition,
                 response = CSQConversionOutliersRM$NauseaWeight,
                 fun = mean,
                 xlab = "Timepoint",
                 ylab = "Score",
                 type= "b",
                 col=c("red","blue"),
                 pch=c(19,17,15),
                 trace.label = "Nausea",
                 fixed=T,
                 leg.bty ="o")
```




```
interaction.plot(x.factor =  CSQConversionOutliersRM$Timepoint,
                 trace.factor = CSQConversionOutliersRM$Condition,
                 response = CSQConversionOutliersRM$OculomotorWeight,
                 fun = mean,
                 xlab = "Timepoint",
                 ylab = "Score",
                 type= "b",
                 col=c("red","blue"),
                 pch=c(19,17,15),
                 trace.label = "Oculomotor",
                 fixed=T,
                 leg.bty ="o")
```




```
interaction.plot(x.factor =  CSQConversionOutliersRM$Timepoint,
                 trace.factor = CSQConversionOutliersRM$Condition,
                 response = CSQConversionOutliersRM$DisorientationWeight,
                 fun = mean,
                 xlab = "Timepoint",
                 ylab = "Score",
                 type= "b",
                 col=c("red","blue"),
                 pch=c(19,17,15),
                 trace.label = "Disorientation",
                 fixed=T,
                 leg.bty ="o")
```




```
CSQModel2 = lm(TotalCSQScore ~ Timepoint + Condition + Timepoint:Condition,
               data = CSQConversionOutliersRM)

Anova(CSQModel2,
      type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: TotalCSQScore
##                      Sum Sq  Df F value Pr(>F)
## Timepoint              75.8   2  0.7980 0.4529
## Condition              17.5   1  0.3679 0.5455
## Timepoint:Condition   123.6   2  1.3021 0.2763
## Residuals            4937.9 104
```

```
OcuModel2 = lm(OculomotorWeight ~ Timepoint + Condition + Timepoint:Condition,
               data = CSQConversionOutliersRM)

Anova(OcuModel2,
      type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: OculomotorWeight
```

```
##                      Sum Sq  Df F value Pr(>F)
## Timepoint             273.2   2  1.6002 0.2068
## Condition              64.7   1  0.7578 0.3860
## Timepoint:Condition   118.0   2  0.6911 0.5033
## Residuals            8879.2 104
```

```
NauModel2 = lm(NauseaWeight ~ Timepoint + Condition + Timepoint:Condition,
               data = CSQConversionOutliersRM)

Anova(NauModel2,
      type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: NauseaWeight
##                      Sum Sq  Df F value Pr(>F)
## Timepoint             117.4   2  1.3680 0.2591
## Condition              11.5   1  0.2692 0.6050
## Timepoint:Condition    10.9   2  0.1274 0.8805
## Residuals            4461.4 104
```

```
DisModel2 = lm(DisorientationWeight ~ Timepoint + Condition + Timepoint:Condition,
               data = CSQConversionOutliersRM)

Anova(DisModel2,
      type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: DisorientationWeight
##                      Sum Sq  Df F value  Pr(>F)
## Timepoint             176.2   2  1.7888 0.17226
## Condition              33.0   1  0.6694 0.41513
## Timepoint:Condition   456.2   2  4.6307 0.01184 *
## Residuals            5122.5 104
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

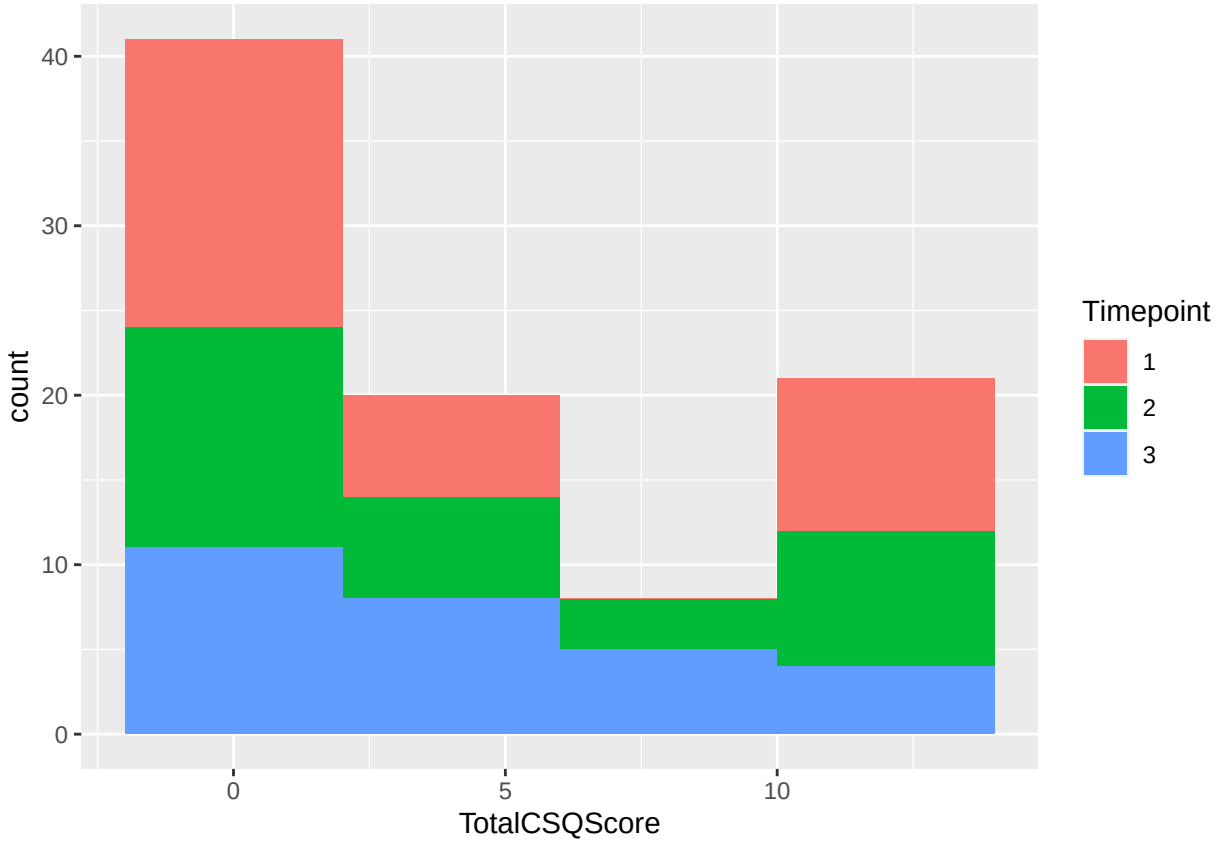
```
Dis2marginal = emmeans(DisModel2, ~Timepoint:Condition)
pairs(Dis2marginal, adjust="tukey")
```

```
##  contrast            estimate   SE  df t.ratio p.value
##  1 Global - 2 Global   -2.030 2.41 104  -0.842  0.9589
##  1 Global - 3 Global    1.791 2.44 104   0.733  0.9774
##  1 Global - 1 Local     1.914 2.35 104   0.813  0.9645
##  1 Global - 2 Local    -0.321 2.38 104  -0.135  1.0000
##  1 Global - 3 Local    -5.046 2.35 104  -2.144  0.2735
##  2 Global - 3 Global    3.821 2.37 104   1.610  0.5940
##  2 Global - 1 Local     3.944 2.28 104   1.730  0.5156
##  2 Global - 2 Local     1.709 2.31 104   0.741  0.9763
##  2 Global - 3 Local    -3.016 2.28 104  -1.323  0.7718
##  3 Global - 1 Local     0.123 2.32 104   0.053  1.0000
```

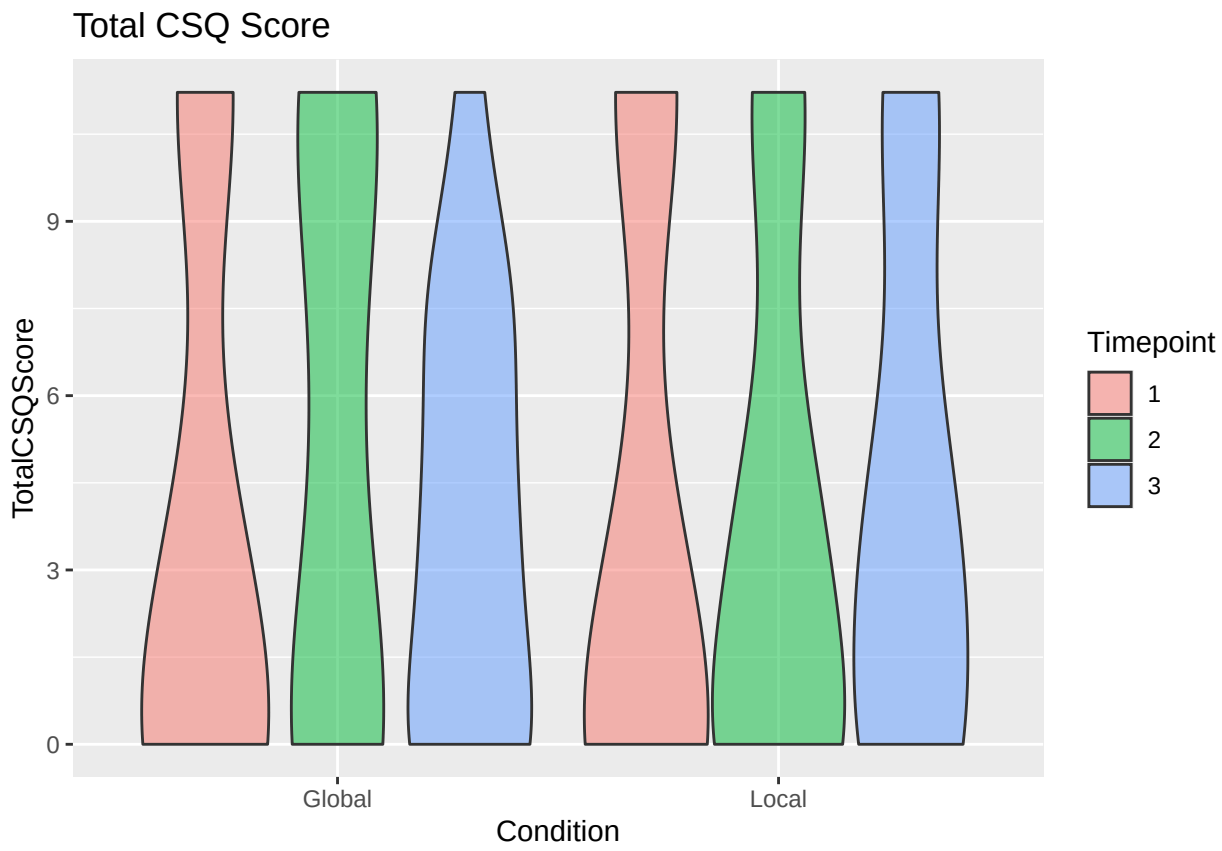
```
##  3 Global - 2 Local    -2.112 2.34 104  -0.901  0.9454
##  3 Global - 3 Local    -6.837 2.32 104  -2.953  0.0438
##  1 Local - 2 Local     -2.235 2.25 104  -0.994  0.9190
##  1 Local - 3 Local     -6.960 2.22 104  -3.136  0.0263
##  2 Local - 3 Local     -4.725 2.25 104  -2.102  0.2945
##
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
CSQConversionOutliersRM75 <- CSQConversion %>%
#Remove outlier participants
subset(TotalCSQScore < quantile(CSQConversion$TotalCSQScore, probs = .75), na.rm = F)


# subset(ID != "P07") %>%
# subset(ID != "P13") %>%
# subset(ID != "P16") %>%
# subset(ID != "P19") %>%
# subset(ID != "P21")

# CSQDataZScores <- CSQdata %>%
#   group_by(Condition, Timepoint)


#z_scores <- as.data.frame(sapply(df, function(df) (abs(df-mean(df))/sd(df))))
```

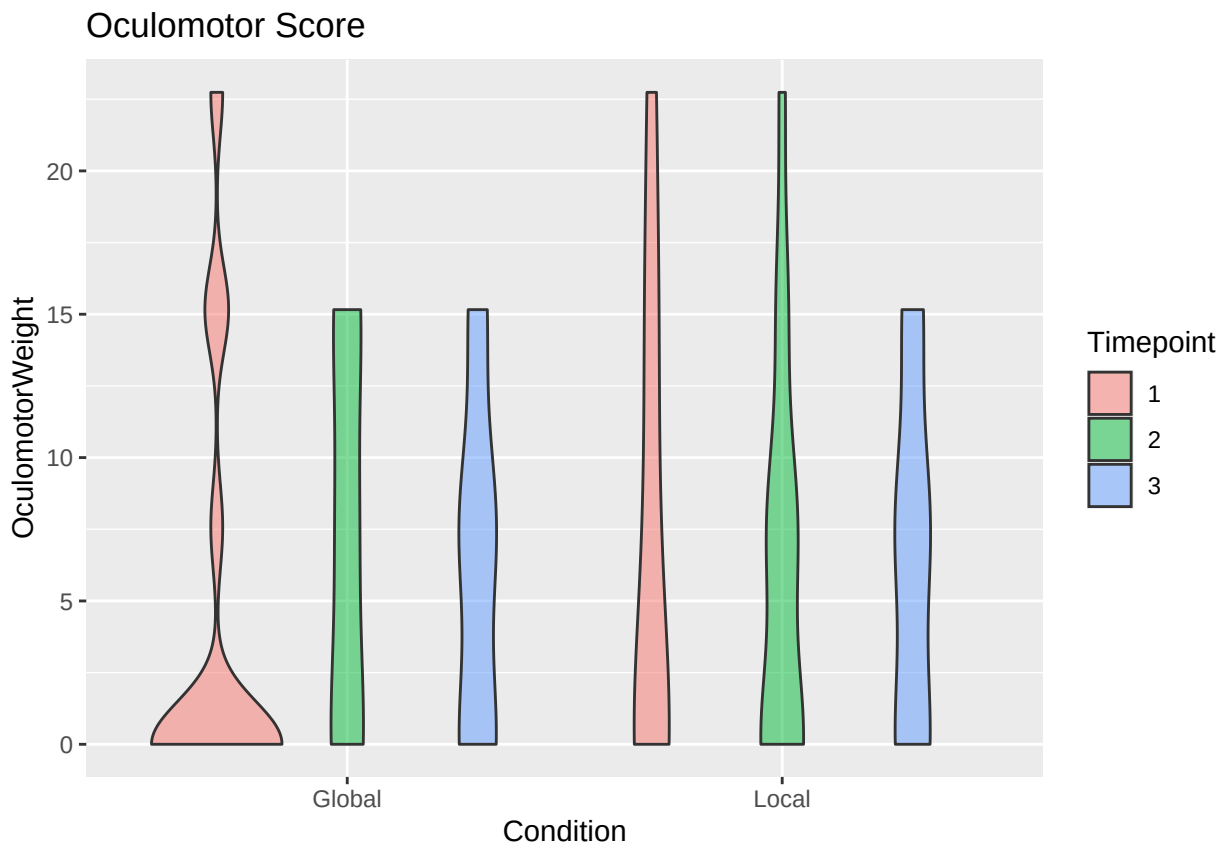
```
ggplot(CSQConversionOutliersRM75, aes(x = TotalCSQScore, fill = Timepoint ))+
  geom_histogram(binwidth = 4)
```



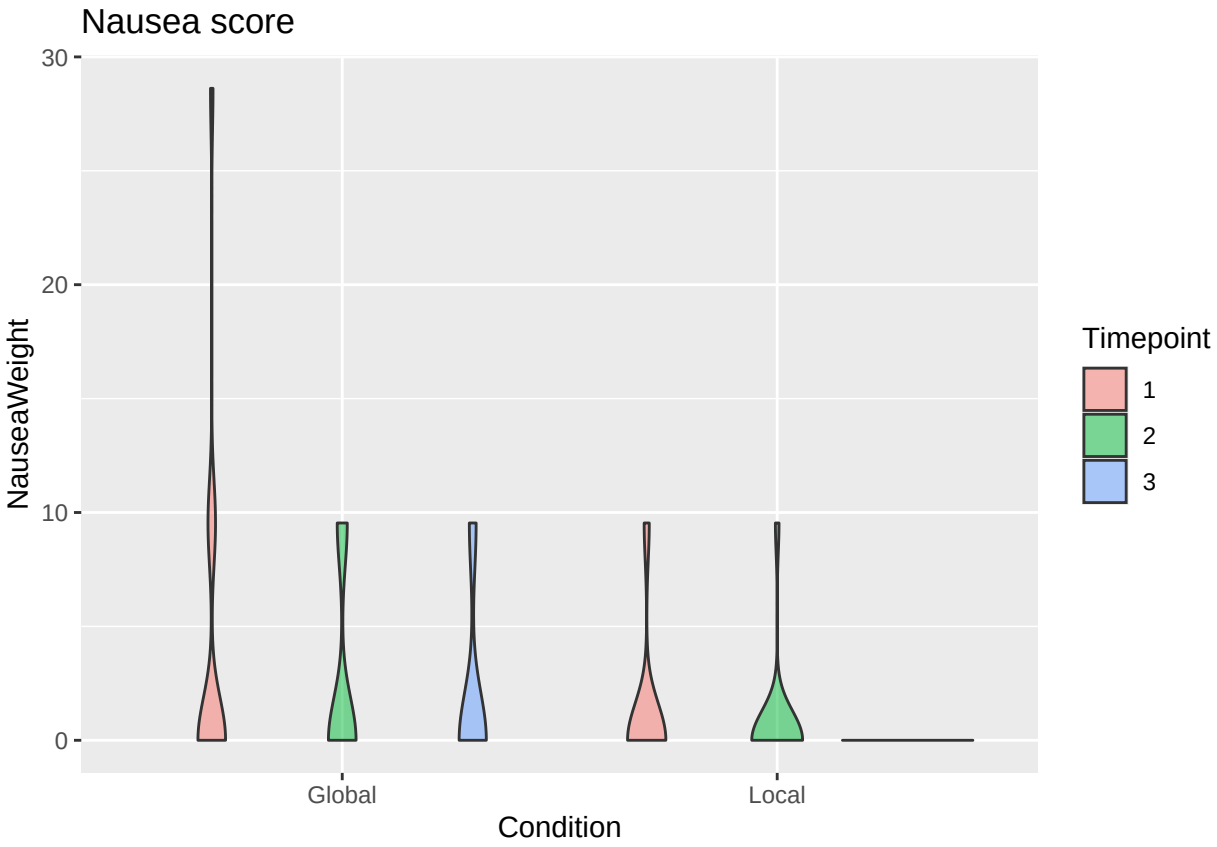

```
ggplot(CSQConversionOutliersRM75, aes(Condition, TotalCSQScore, fill = Timepoint ))+
  geom_violin(alpha = 0.5)+
  labs(title = "Total CSQ Score")
```



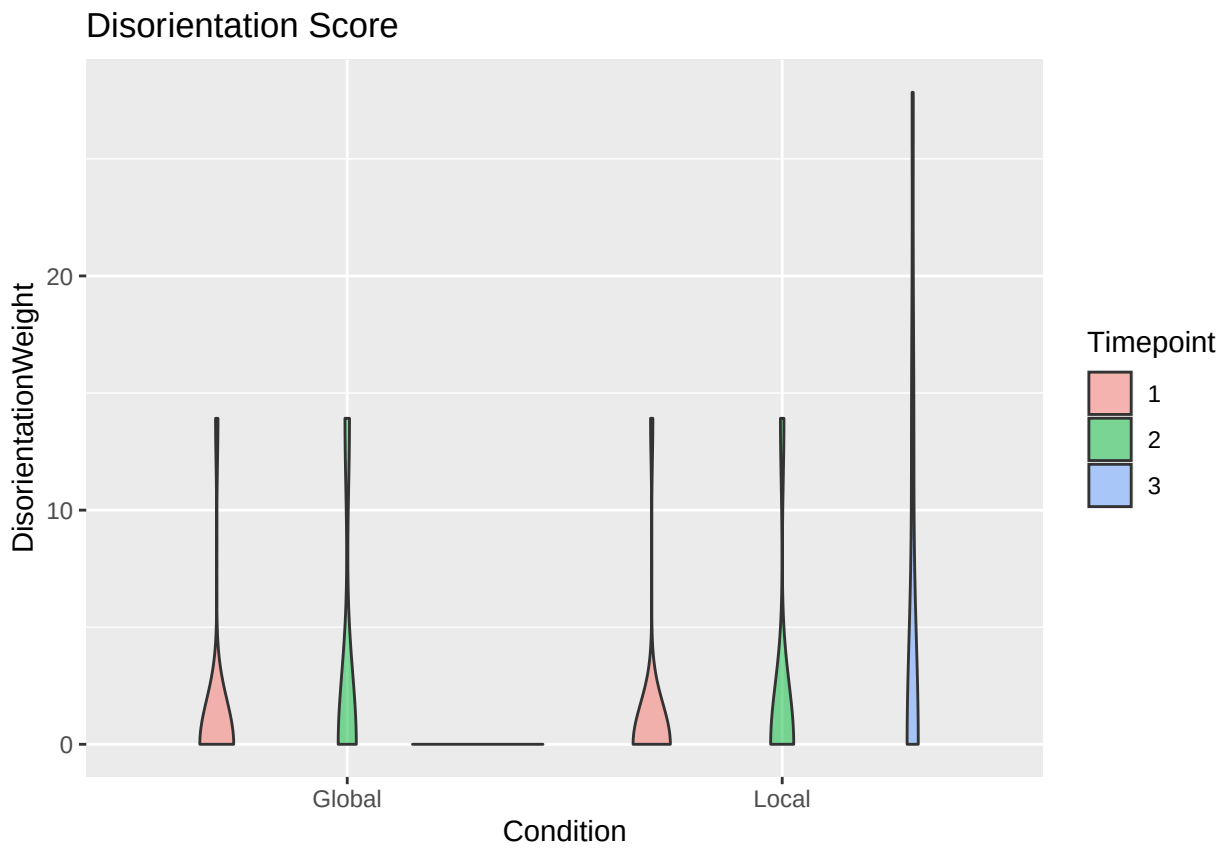

```
ggplot(CSQConversionOutliersRM75, aes(Condition, OculomotorWeight, fill = Timepoint ))+
  geom_violin(alpha = 0.5) +
  labs(title = "Oculomotor Score")
```



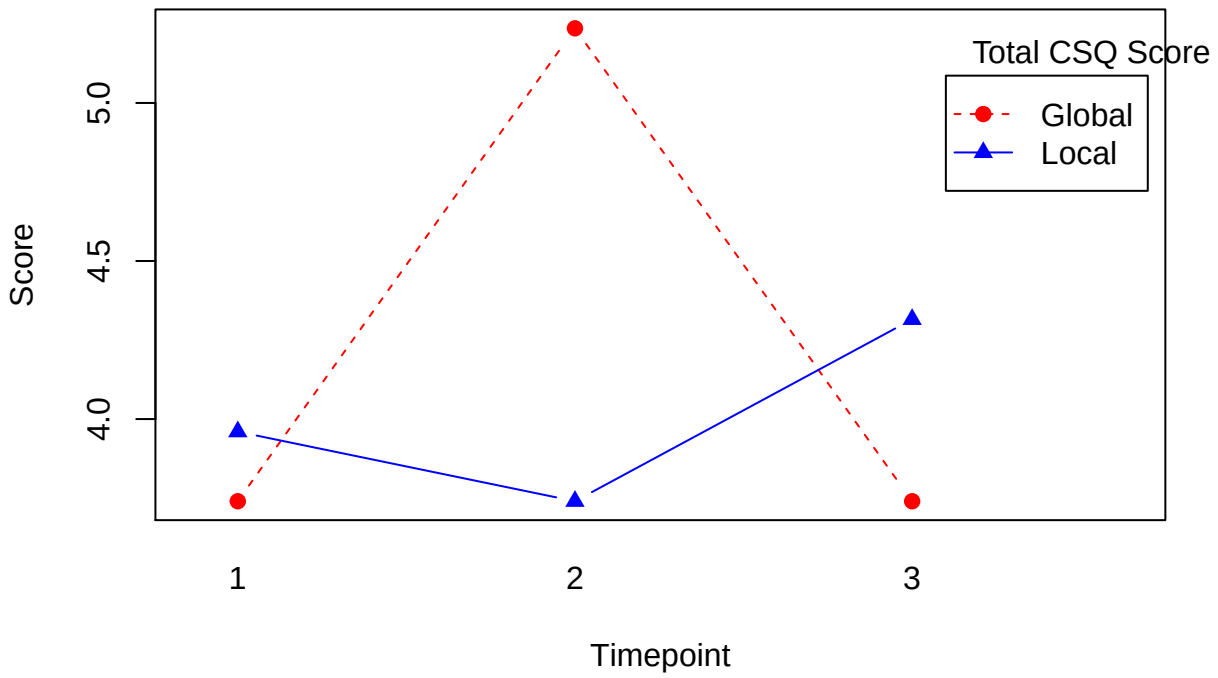

```
ggplot(CSQConversionOutliersRM75, aes(Condition, NauseaWeight, fill = Timepoint ))+
  geom_violin(alpha = 0.5) +
  labs(title = "Nausea score")
```



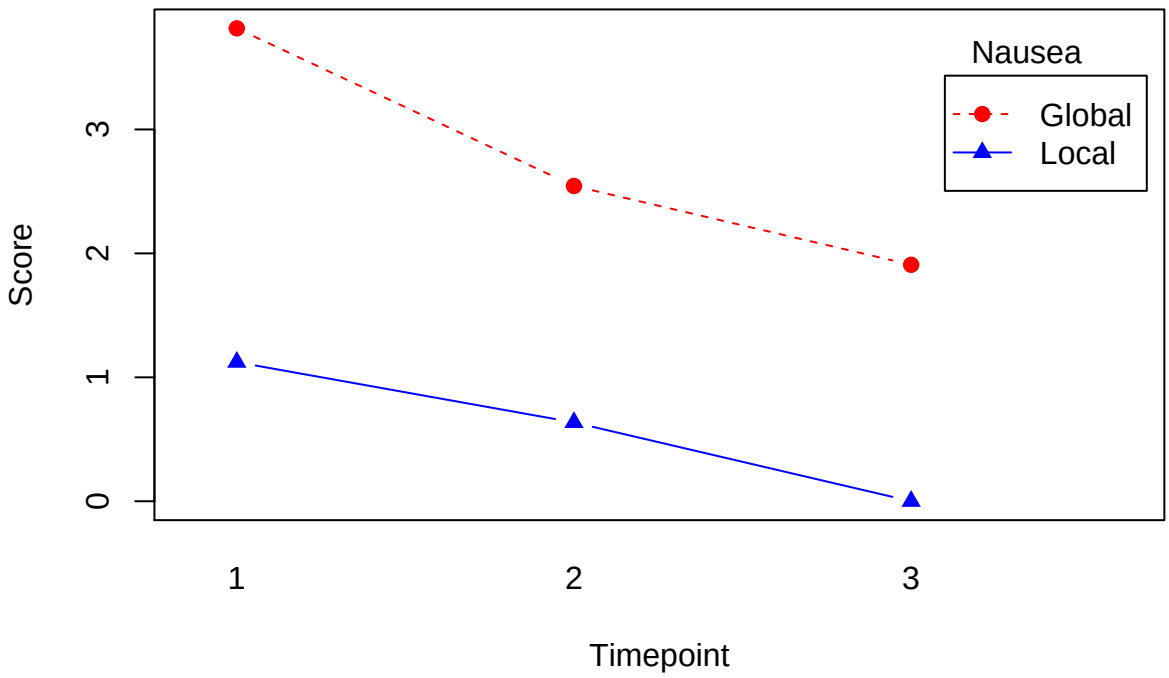

```
ggplot(CSQConversionOutliersRM75, aes(Condition, DisorientationWeight, fill = Timepoint ))+
  geom_violin(alpha = 0.5)+
  labs(title = "Disorientation Score")
```



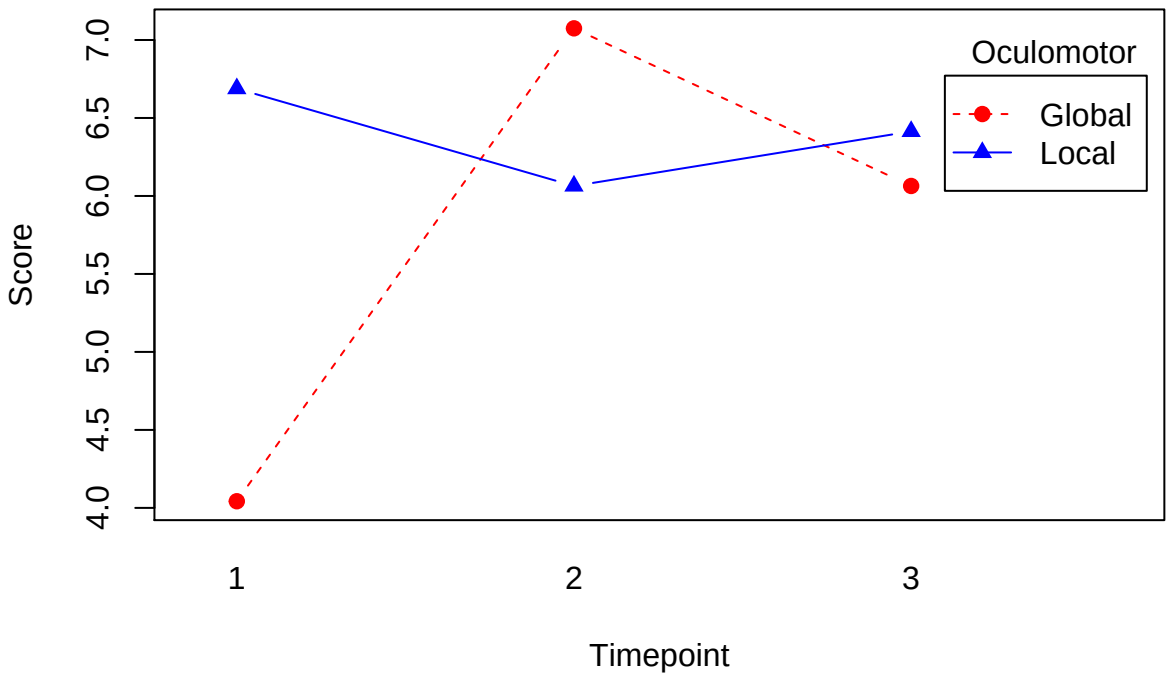

```
interaction.plot(x.factor =  CSQConversionOutliersRM75$Timepoint,
                 trace.factor = CSQConversionOutliersRM75$Condition,
                 response = CSQConversionOutliersRM75$TotalCSQScore,
                 fun = mean,
                 xlab = "Timepoint",
                 ylab = "Score",
                 type= "b",
                 col=c("red","blue"),
                 pch=c(19,17,15),
                 trace.label = "Total CSQ Score",
                 fixed=T,
                 leg.bty ="o")
```



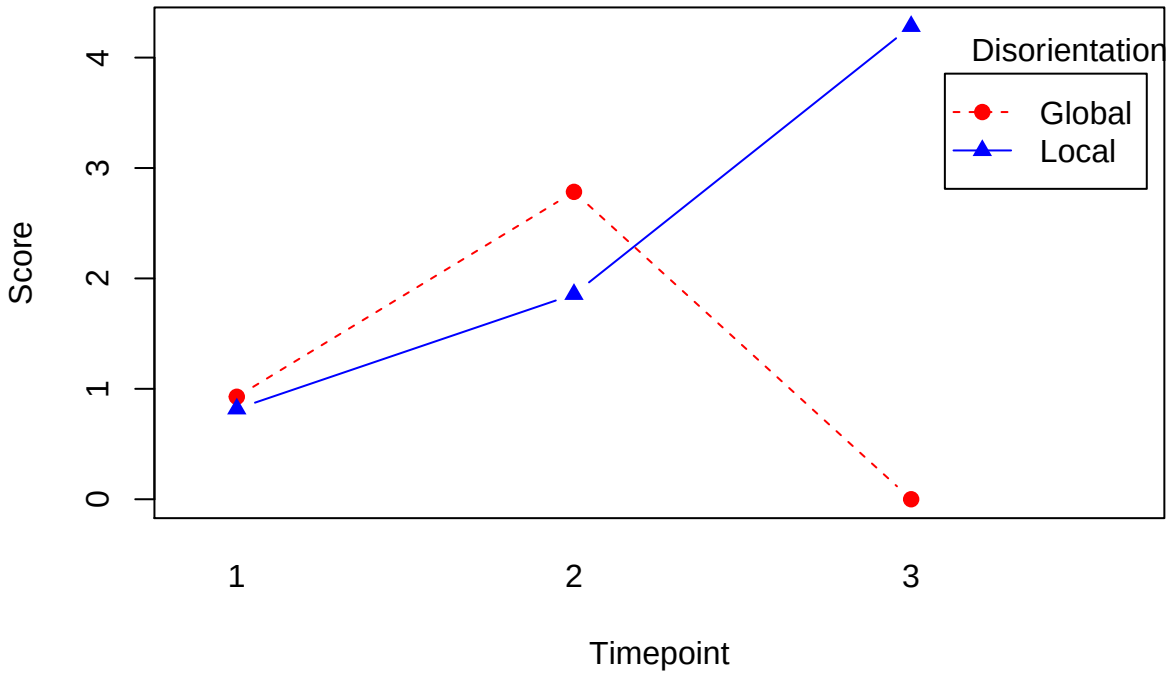

```
interaction.plot(x.factor =  CSQConversionOutliersRM75$Timepoint,
                 trace.factor = CSQConversionOutliersRM75$Condition,
                 response = CSQConversionOutliersRM75$NauseaWeight,
                 fun = mean,
                 xlab = "Timepoint",
                 ylab = "Score",
                 type= "b",
                 col=c("red","blue"),
                 pch=c(19,17,15),
                 trace.label = "Nausea",
                 fixed=T,
                 leg.bty ="o")
```




```
interaction.plot(x.factor =  CSQConversionOutliersRM75$Timepoint,
                 trace.factor = CSQConversionOutliersRM75$Condition,
                 response = CSQConversionOutliersRM75$OculomotorWeight,
                 fun = mean,
                 xlab = "Timepoint",
                 ylab = "Score",
                 type= "b",
                 col=c("red","blue"),
                 pch=c(19,17,15),
                 trace.label = "Oculomotor",
                 fixed=T,
                 leg.bty ="o")
```




```
interaction.plot(x.factor =  CSQConversionOutliersRM75$Timepoint,
                 trace.factor = CSQConversionOutliersRM75$Condition,
                 response = CSQConversionOutliersRM75$DisorientationWeight,
                 fun = mean,
                 xlab = "Timepoint",
                 ylab = "Score",
                 type= "b",
                 col=c("red","blue"),
                 pch=c(19,17,15),
                 trace.label = "Disorientation",
                 fixed=T,
                 leg.bty ="o")
```




```
CSQModel3 = lm(TotalCSQScore ~ Timepoint + Condition + Timepoint:Condition,
               data = CSQConversionOutliersRM75)

Anova(CSQModel3,
      type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: TotalCSQScore
##                       Sum Sq Df F value Pr(>F)
## Timepoint               6.56  2  0.1513 0.8598
## Condition               1.33  1  0.0613 0.8051
## Timepoint:Condition    18.15  2  0.4184 0.6595
## Residuals            1821.65 84
```

```
OcuModel3 = lm(OculomotorWeight ~ Timepoint + Condition + Timepoint:Condition,
               data = CSQConversionOutliersRM75)

Anova(OcuModel3,
      type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: OculomotorWeight
```

```
##                      Sum Sq Df F value Pr(>F)
## Timepoint             21.6  2  0.2170 0.8054
## Condition             11.3  1  0.2270 0.6350
## Timepoint:Condition   53.0  2  0.5309 0.5900
## Residuals           4189.9 84
```

```
NauModel3 = lm(NauseaWeight ~ Timepoint + Condition + Timepoint:Condition,
               data = CSQConversionOutliersRM75)

Anova(NauModel3,
      type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: NauseaWeight
##                      Sum Sq Df F value  Pr(>F)
## Timepoint             34.42  2  0.9010 0.41006
## Condition            107.30  1  5.6171 0.02008 *
## Timepoint:Condition    3.17  2  0.0830 0.92042
## Residuals           1604.66 84
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
DisModel3 = lm(DisorientationWeight ~ Timepoint + Condition + Timepoint:Condition,
               data = CSQConversionOutliersRM75)

Anova(DisModel3,
      type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: DisorientationWeight
##                      Sum Sq Df F value Pr(>F)
## Timepoint             38.10  2  0.7664 0.4679
## Condition             21.57  1  0.8678 0.3542
## Timepoint:Condition  112.74  2  2.2675 0.1099
## Residuals           2088.23 84
```

```
Nau3marginal = emmeans(NauModel3, ~Timepoint:Condition)
pairs(Nau3marginal, adjust="tukey")
```

```
##  contrast            estimate   SE df t.ratio p.value
##  1 Global - 2 Global    1.272 1.60 84   0.797  0.9673
##  1 Global - 3 Global    1.908 1.60 84   1.196  0.8378
##  1 Global - 1 Local     2.694 1.55 84   1.740  0.5099
##  1 Global - 2 Local     3.180 1.60 84   1.993  0.3553
##  1 Global - 3 Local     3.816 1.66 84   2.304  0.2040
##  2 Global - 3 Global    0.636 1.60 84   0.399  0.9987
##  2 Global - 1 Local     1.422 1.55 84   0.918  0.9408
##  2 Global - 2 Local     1.908 1.60 84   1.196  0.8378
##  2 Global - 3 Local     2.544 1.66 84   1.536  0.6422
##  3 Global - 1 Local     0.786 1.55 84   0.507  0.9958
```

```
##  3 Global - 2 Local     1.272 1.60 84   0.797  0.9673
##  3 Global - 3 Local     1.908 1.66 84   1.152  0.8579
##  1 Local - 2 Local      0.486 1.55 84   0.314  0.9996
##  1 Local - 3 Local      1.122 1.61 84   0.697  0.9818
##  2 Local - 3 Local      0.636 1.66 84   0.384  0.9989
##
## P value adjustment: tukey method for comparing a family of 6 estimates
```