Virtual Patient Questionnaire Analysis

Han Lie

December 2018

Introduction

This document gives insight in the analysis that was done on the questionnaire data which was collected in an experiment done with a virtual patient for shared decision making. This questionnaire consisted of three subquestionnaires:

- Perceived Influence of Learning Outcomes (PILO) - measures a subject's knowledge on SDM, SDM skills, attitude towards virtual patients and self-efficacy through self assessment.
- Dialogue Experience Questionnaire (DEQ) - measures the dialogue experience a user has with the virtual patient. It takes into account two dimensions: flow and interaction, each divided into several subdimensions. Flow tries to measure whether the dialogue experience progresses in a proper manner. It does so by measuring dialogue speed, interruption, correctness locally and correctness globally. Interaction tries to measure the immersion the user feels throughout the interaction with the virtual patient. It uses involvement, discussion satisfaction and reality to achieve this.
- System Usability Scale (SUS) - measures the attitude towards a system. It covers the effectiveness and efficiency of a system and a user's satisfaction with the system.

Read in the questionnaire data:

```
questionnaire <- read.csv(file=paste("/Users/fooyonghan/Downloads/final-analysis/",
  "questionnaire-analysis/questionnaire-results-final.csv", sep=""), header=TRUE, sep=";")
```

PILO

Libraries used:

```
library(ggplot2)        # plotting & data
library(dplyr)          # data manipulation
library(tidyr)          # data re-shaping
library(magrittr)       # pipe operator
library(gridExtra)
```

Create a bar plot of the PILO questionnaire showing the mean and standard deviation of each question.

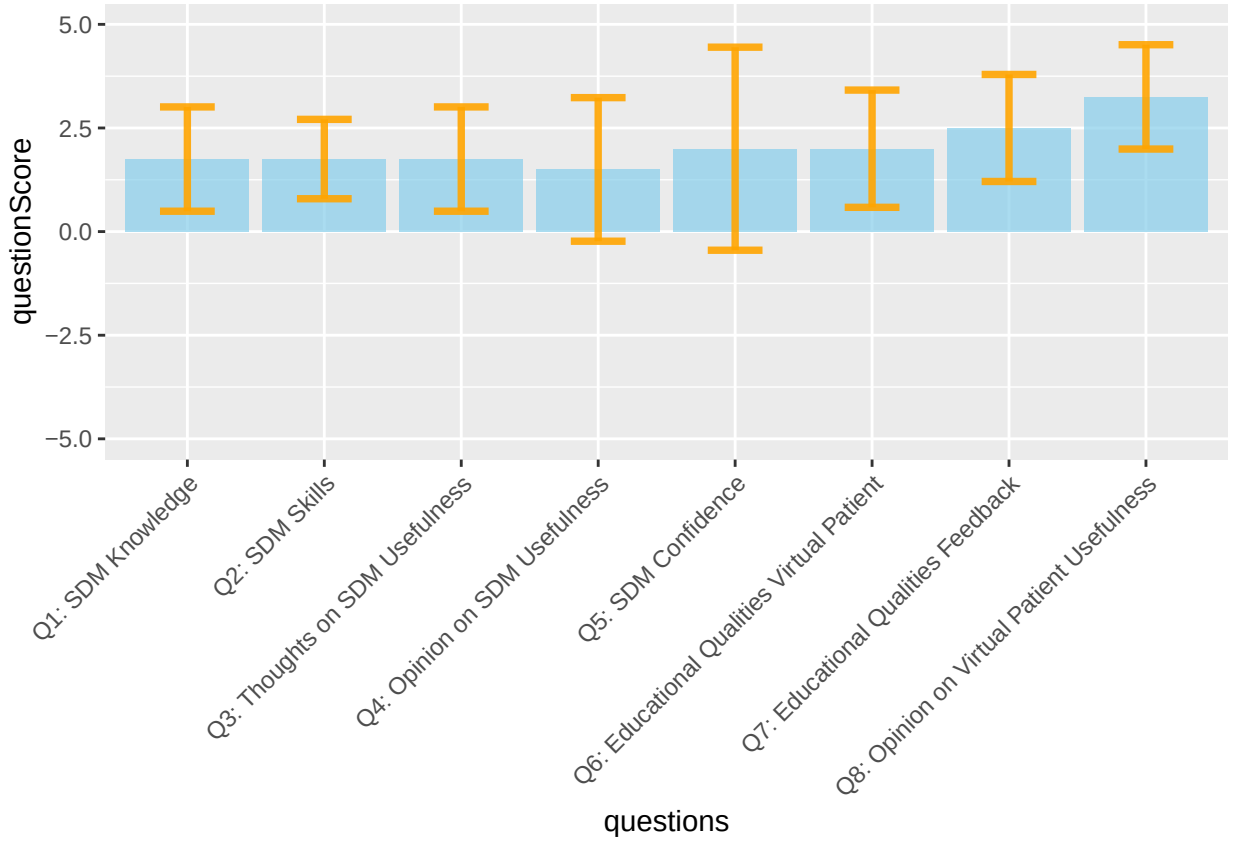
```
experience <- questionnaire[c(0:4),c(10:17)]

expMean <- colMeans(experience)
expSds <- sapply(experience, sd, na.rm = TRUE)
# questions <- c("Q1","Q2","Q3","Q4","Q5","Q6","Q7","Q8")
questions <- c(
  "Q1: SDM Knowledge",
  "Q2: SDM Skills",
  "Q3: Thoughts on SDM Usefulness",
  "Q4: Opinion on SDM Usefulness",
  "Q5: SDM Confidence",
  "Q6: Educational Qualities Virtual Patient",
  "Q7: Educational Qualities Feedback",
```

```
  "Q8: Opinion on Virtual Patient Usefulness"
)

PILOFrame <- data.frame("questions" = questions, "expMean" = expMean, "expSds" = expSds)

ggplot(PILOFrame) +
  geom_bar( aes(
    x=PILOFrame$questions,
    y=PILOFrame$expMean),
    stat="identity",
    fill="skyblue", alpha=0.7) +
  geom_errorbar( aes(
    x=PILOFrame$questions,
    ymin=PILOFrame$expMean-PILOFrame$expSds,
    ymax=PILOFrame$expMean+PILOFrame$expSds),
    width=0.4,
    colour="orange",
    alpha=0.9,
    size=1.3) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  xlab("questions") +
  ylab("questionScore") +
  expand_limits(y=c(-5,5))
```



t-test of PILO questionnaire

```r
for (i in 1:length(experience)){
  print(t.test(experience[[i]], mu = 0, alternative = "two.sided"))
}
```

```
##
##  One Sample t-test
##
## data:  experience[[i]]
## t = 2.7815, df = 3, p-value = 0.0689
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.2522452  3.7522452
## sample estimates:
## mean of x
##      1.75
##
##
##  One Sample t-test
##
## data:  experience[[i]]
## t = 3.6556, df = 3, p-value = 0.03535
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.2265198 3.2734802
## sample estimates:
## mean of x
##      1.75
##
##
##  One Sample t-test
##
## data:  experience[[i]]
## t = 2.7815, df = 3, p-value = 0.0689
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.2522452  3.7522452
## sample estimates:
## mean of x
##      1.75
##
##
##  One Sample t-test
##
## data:  experience[[i]]
## t = 1.7321, df = 3, p-value = 0.1817
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -1.256079  4.256079
## sample estimates:
## mean of x
##       1.5
##
##
##  One Sample t-test
```

```
##
## data:  experience[[i]]
## t = 1.633, df = 3, p-value = 0.201
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -1.897685  5.897685
## sample estimates:
## mean of x
##         2
##
##
##  One Sample t-test
##
## data:  experience[[i]]
## t = 2.8284, df = 3, p-value = 0.06628
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.2503294  4.2503294
## sample estimates:
## mean of x
##         2
##
##
##  One Sample t-test
##
## data:  experience[[i]]
## t = 3.873, df = 3, p-value = 0.03047
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.4457397 4.5542603
## sample estimates:
## mean of x
##       2.5
##
##
##  One Sample t-test
##
## data:  experience[[i]]
## t = 5.1657, df = 3, p-value = 0.01407
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  1.247755 5.252245
## sample estimates:
## mean of x
##      3.25
```

DEQ

Get Flow and Interaction subdimension data from the questionnaire:

```
dimensions <- questionnaire[c(0:4),c(18:49)]
dimMean <- colMeans(dimensions)
dimSds <- sapply(dimensions, sd, na.rm = TRUE)
```

```r
# Flow
dialogueSpeedMean <- mean(dimMean[c(0:4)])
dialogueSpeedSd <- mean(dimSds[c(0:4)])
interruptionMean <- mean(dimMean[c(5:9)])
interruptionSd <- mean(dimSds[c(5:9)])
correctnessLocallyMean <- mean(dimMean[c(10:13)])
correctnessLocallySd <- mean(dimSds[c(10:13)])
correctnessGloballyMean <- mean(dimMean[c(14:17)])
correctnessGloballySd <- mean(dimSds[c(14:17)])

# Interaction
involvementMean <- mean(dimMean[c(18:22)])
involvementSd <- mean(dimSds[c(18:22)])
discussionSatisfactionMean <- mean(dimMean[c(23:27)])
discussionSatisfactionSd <- mean(dimSds[c(23:27)])
realityMean <- mean(dimMean[c(28:32)])
realitySd <- mean(dimSds[c(28:32)])
```

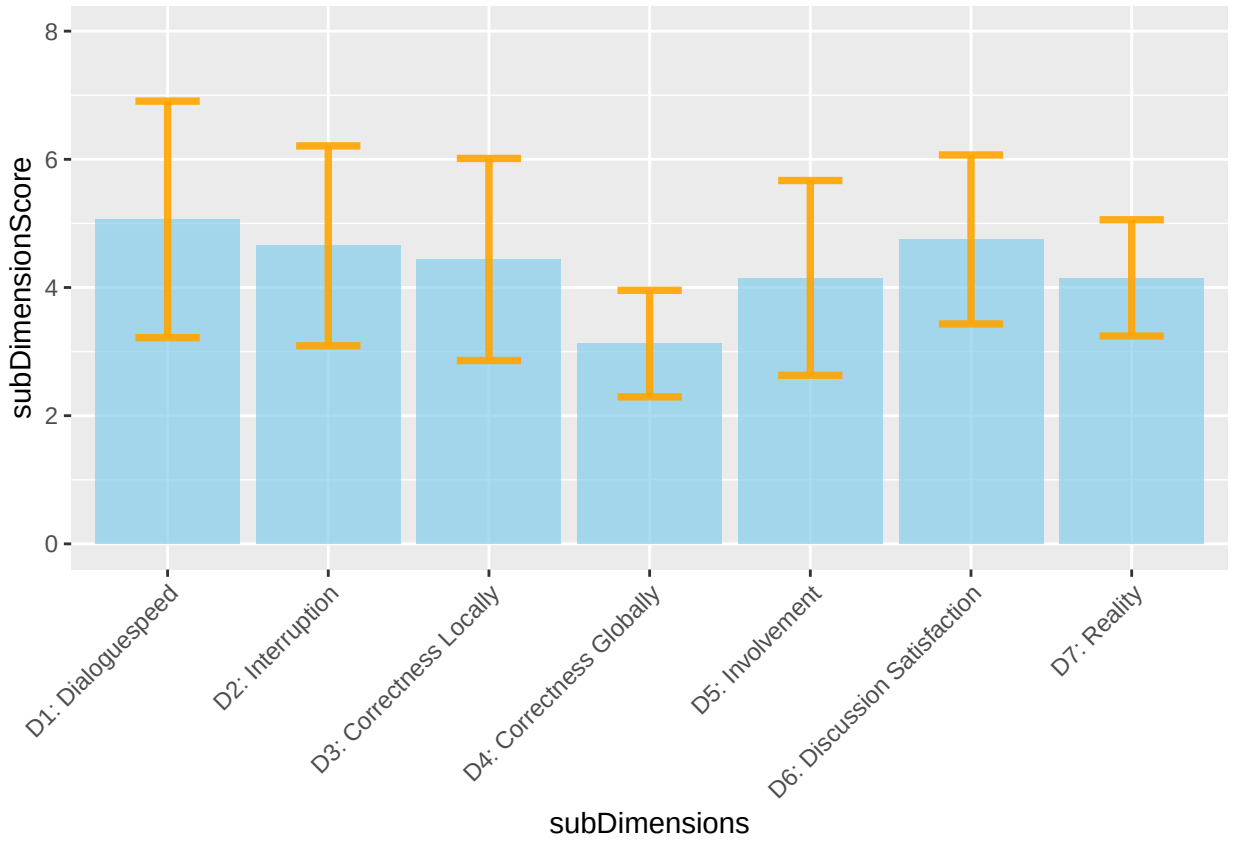
Calculate means and standard deviations of subdimensions of Flow and Interaction:

```r
subDimensionMeans <- c(
  dialogueSpeedMean,
  interruptionMean,
  correctnessLocallyMean,
  correctnessGloballyMean,
  involvementMean,
  discussionSatisfactionMean,
  realityMean
)
subDimensionSds <- c(
  dialogueSpeedSd,
  interruptionSd,
  correctnessLocallySd,
  correctnessGloballySd,
  involvementSd,
  discussionSatisfactionSd,
  realitySd
)
subDimensionNames <- c(
  "D1: Dialoguespeed",
  "D2: Interruption",
  "D3: Correctness Locally",
  "D4: Correctness Globally",
  "D5: Involvement",
  "D6: Discussion Satisfaction",
  "D7: Reality"
)

dimFrame <- data.frame(
  "subDimensionNames" = subDimensionNames,
  "subDimensionMeans" = subDimensionMeans,
  "subDimensionSds" = subDimensionSds
)
```

Plot subdimensions for Flow and Interaction:

```
ggplot(dimFrame) +
  geom_bar( aes(
    x=dimFrame$subDimensionNames,
    y=dimFrame$subDimensionMeans),
    stat="identity",
    fill="skyblue",
    alpha=0.7) +
  geom_errorbar( aes(
    x=dimFrame$subDimensionNames,
    ymin=dimFrame$subDimensionMeans-dimFrame$subDimensionSds,
    ymax=dimFrame$subDimensionMeans+dimFrame$subDimensionSds),
    width=0.4,
    colour="orange",
    alpha=0.9,
    size=1.3) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  xlab("subDimensions") +
  ylab("subDimensionScore") +
  expand_limits(y=c(0,8))
```



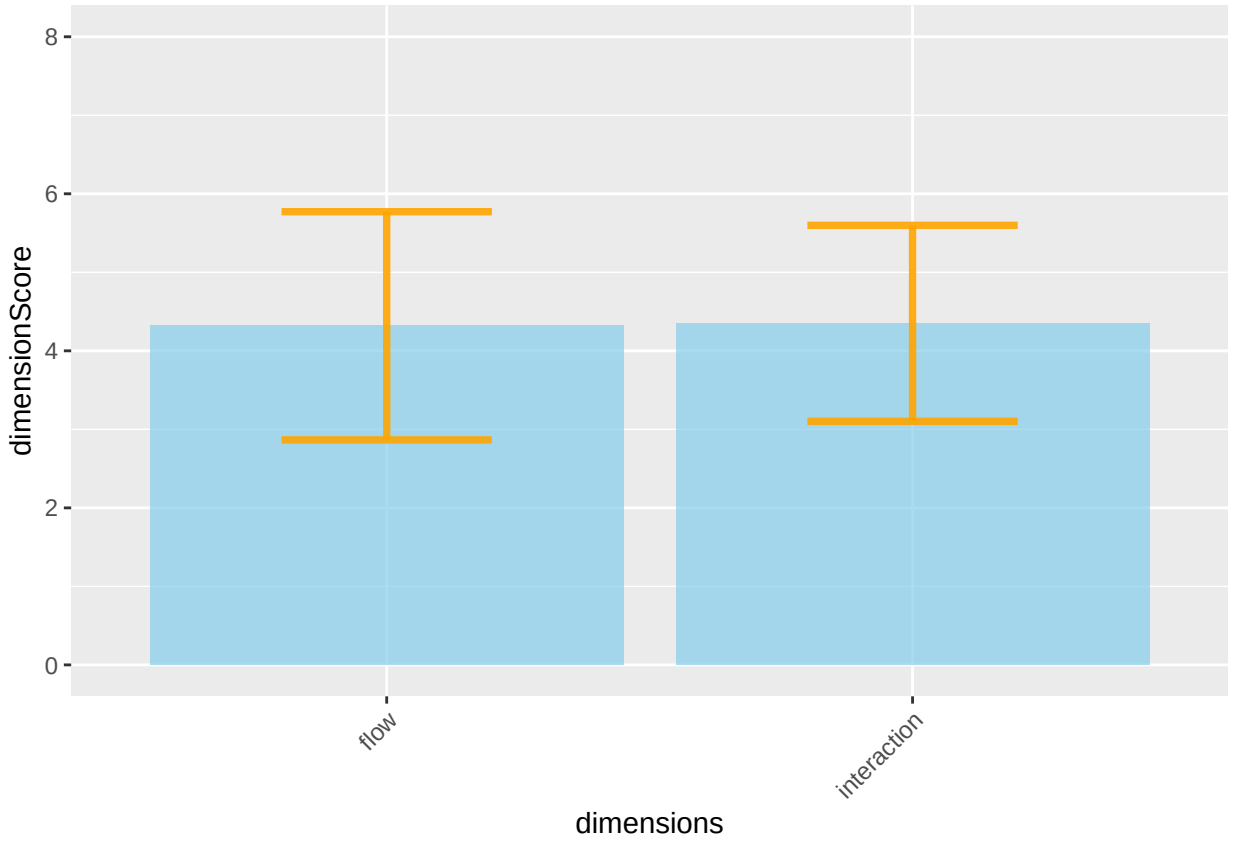
Create bar chart of Flow and Interaction.

```
flowMean <- mean(subDimensionMeans[c(0:4)])
flowSd <- mean(subDimensionSds[c(0:4)])
interactionMean <- mean(subDimensionMeans[c(5:7)])
interactionSd <- mean(subDimensionSds[c(5:7)])
```

```r
FIMeans <- c(flowMean, interactionMean)
FISds <- c(flowSd, interactionSd)
FINames <- c("flow", "interaction")

FIFrame <- data.frame("FINames" = FINames, "FIMeans" = FIMeans, "FISds" = FISds)

ggplot(FIFrame) +
  geom_bar( aes(
    x=FIFrame$FINames,
    y=FIFrame$FIMeans),
    stat="identity",
    fill="skyblue",
    alpha=0.7) +
  geom_errorbar( aes(
    x=FIFrame$FINames,
    ymin=FIFrame$FIMeans-FIFrame$FISds,
    ymax=FIFrame$FIMeans+FIFrame$FISds),
    width=0.4,
    colour="orange",
    alpha=0.9,
    size=1.3) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  xlab("dimensions") +
  ylab("dimensionScore") +
  expand_limits(y=c(0,8))
```



SUS

Calculate mean and std of SUS

```
usability <- questionnaire[c(0:4),c(50:59)]
usabilityMean <- colMeans(usability)
usabilitySds <- sapply(usability, sd, na.rm = TRUE)
sum(usabilityMean) * 2.5
```

```
## [1] 73.75
```