

Hedier - Projet Final

Antoine Hedier

28 décembre 2019

```
\begin{document}
```

The data I'm using is the pre-treated and simplified results of one of my master's thesis experiment, it is a completion experiment with an equation before each uncompleted sentence. Uncompleted sentence ended with a complex Denominal Phrase (NP1 of NP2) and a "qui" which forces the use of a relative clause. Equations were supposed to prime either a high or low attachment (respectively, NP1 or NP2).

Here, I'm reproducing the exploratory statistics I did in my master's thesis. Though I'm not supposed to have previous knowledge of the data, my (secret) goal is to have more high-attachments on the "percep" and "high" condition. High condition is characterized by the equation shown and calculated before completion, it is supposed to prime a high-attachment (NP1). The percep condition is characterized by a perception verb in the uncompleted sequence, perception verbs induce the possibility of pseudo-relatives which are preferred to relative clauses and bypass them for interpretation. The structure of a pseudo-relative is different than a relative clause (the PR is in a sisterhood relation with the NP compared to a embedded relation for relative clause) so we expect no differences between high and low condition within the percep condition.

```
data <- read.csv(file="https://raw.githubusercontent.com/anthdr/Statistique-M2/master/statistique%20descriptive/projet%20final/data.csv", header=TRUE, sep=",")
```

The variable "id" refers to a unique and single participant of this experiment. Condition indicates sentences and condition the declination of this sentence: the first word ("percep" or "stat") refers to the nature of the verb used in the uncompleted sentence and the second word ("high" or "low") refers to the nature of the equation prime. "rep.equation" is the answer of the participant to the equation. "res equation" is the correct (and expected) answer to the equation. "cor.equation" is 1 if the participant answer matches the correct answer, 0 if not. "amount.cor.equation" is the sum of total responses for a given participant "RT.equation" is the time used by the participant to answer. "completion" contains the raw completion of the participant. "nbr.NP1" is the number of the first NP in the complex NP in the uncompleted sentence. "nbr.comp" is the number of the verb used in the participant completion. "attachment" is 1 if the attachment of the relative clause (written by the participant) have a high-attachment, 0 if low. "RT.completion" is the time used by the participant to complete the sentence. "halfsession" is 0 for the first half of the experiment and 1 for the second half. "age" is the age of the participant.

Here is all the packages I will need.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.1
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
## filter, lag
```

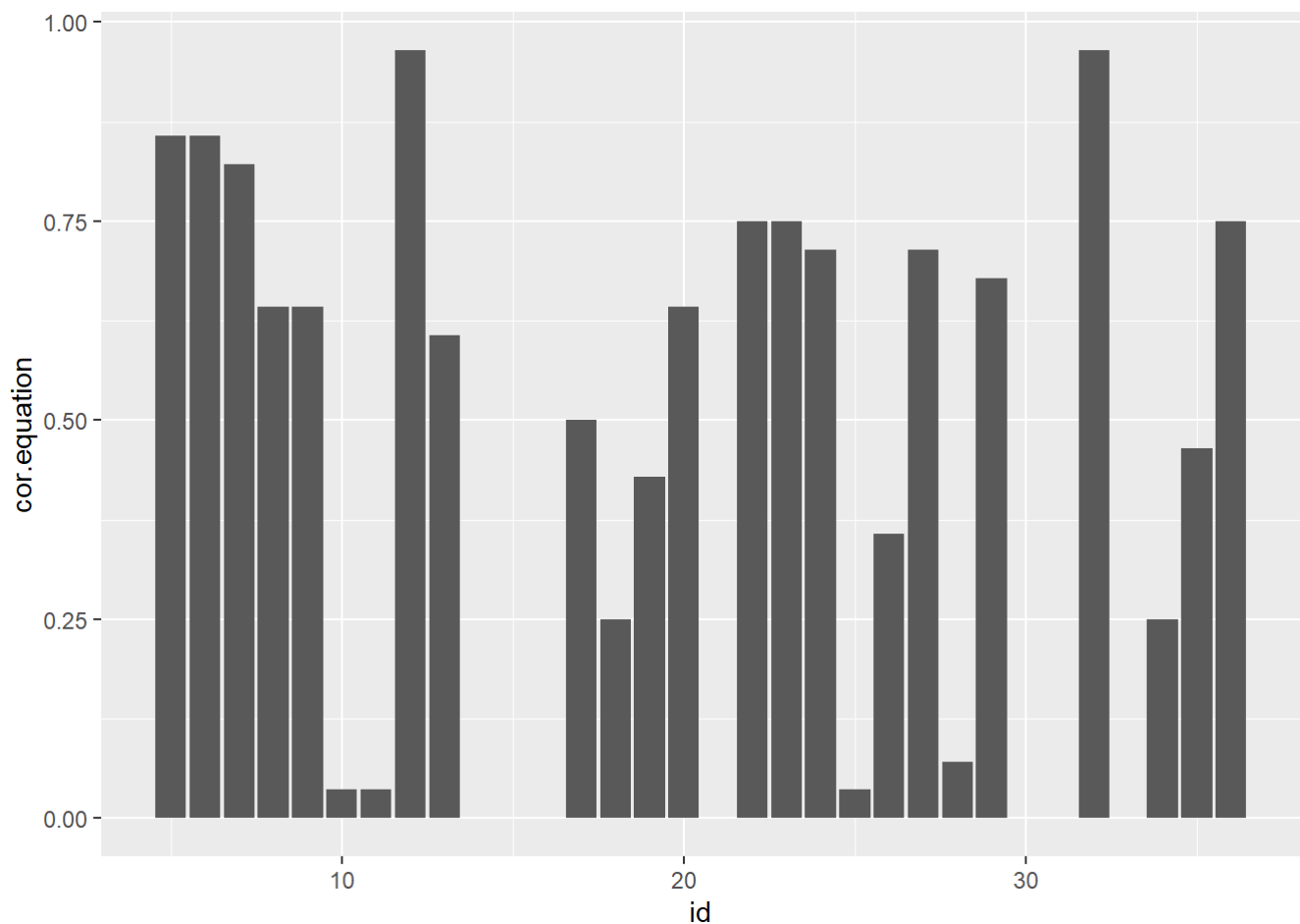
```
## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

I first plot good responses for each participant

```
ggplot(data=data) +
  geom_bar(mapping = aes(x = id, y = cor.equation), stat = "summary")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

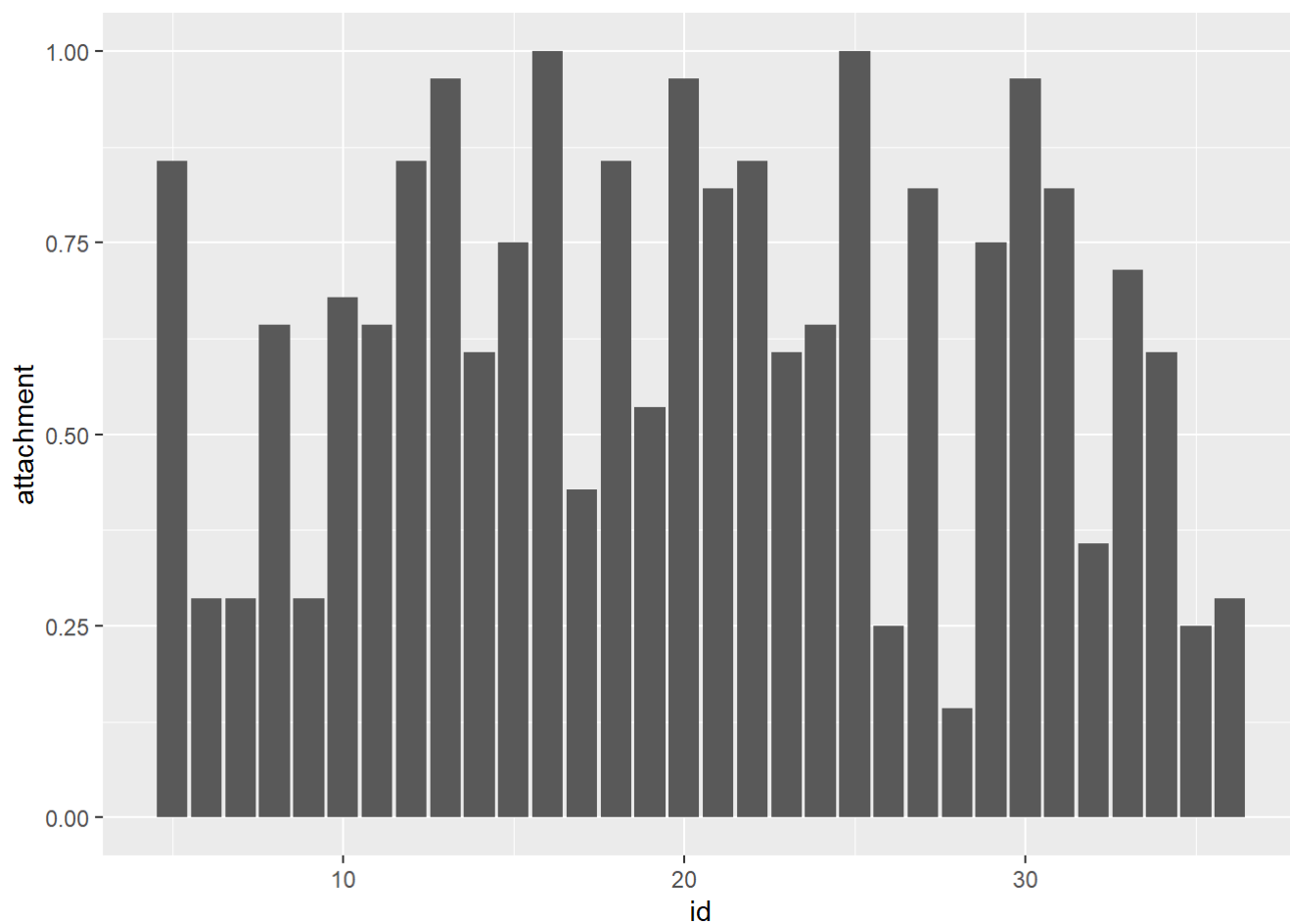


We can see that there is variability in data throughout participants.

Second, I plot attachment for each participant

```
ggplot(data=data) +
  geom_bar(mapping = aes(x = id, y = attachment), stat = "summary")
```

```
## No summary function supplied, defaulting to `mean_se()`
```



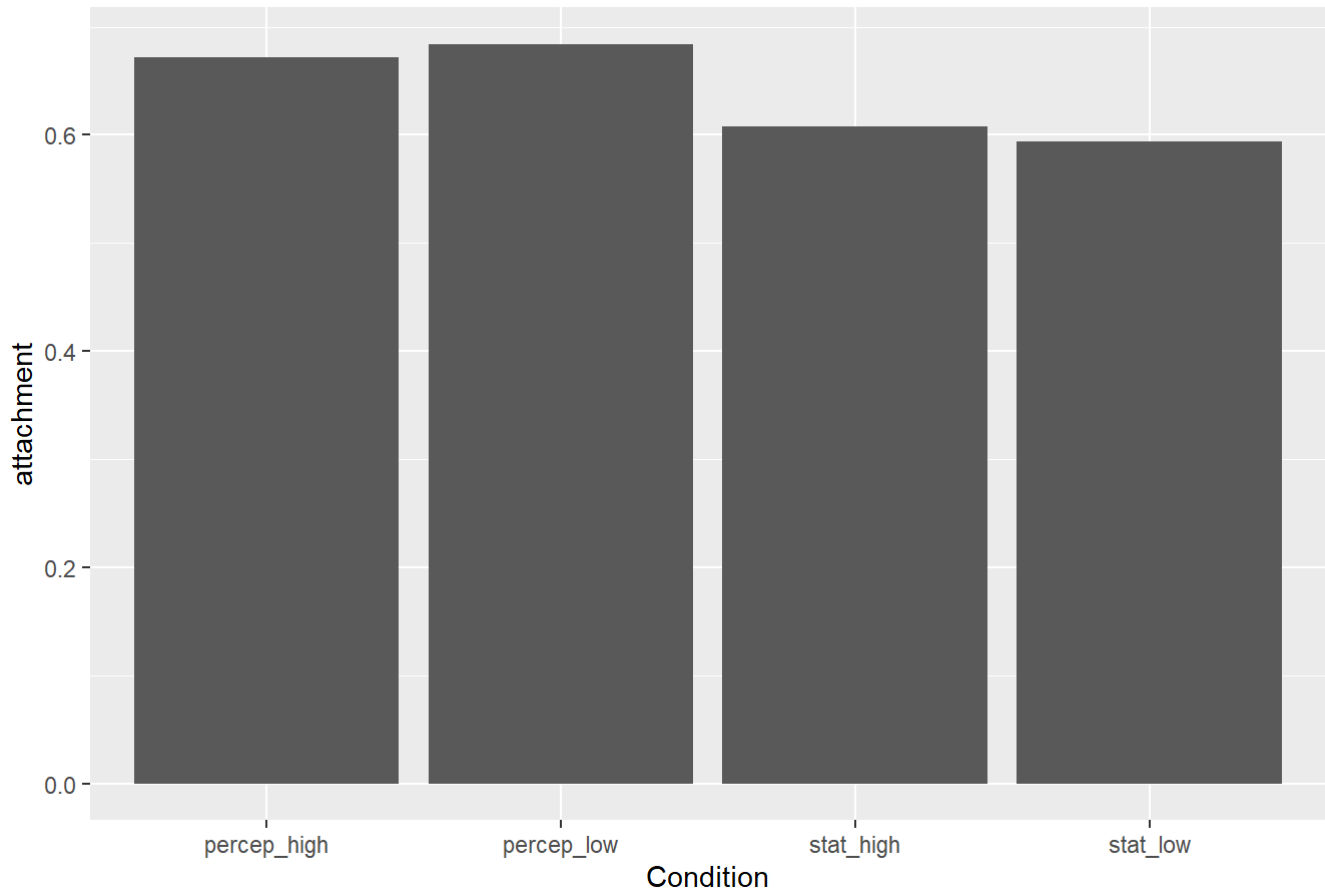
Once again, there is variability in the data.

Then, I plot attachment for each condition

```
ggplot(data=data) +
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +
  ggtitle("Overall")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

Overall



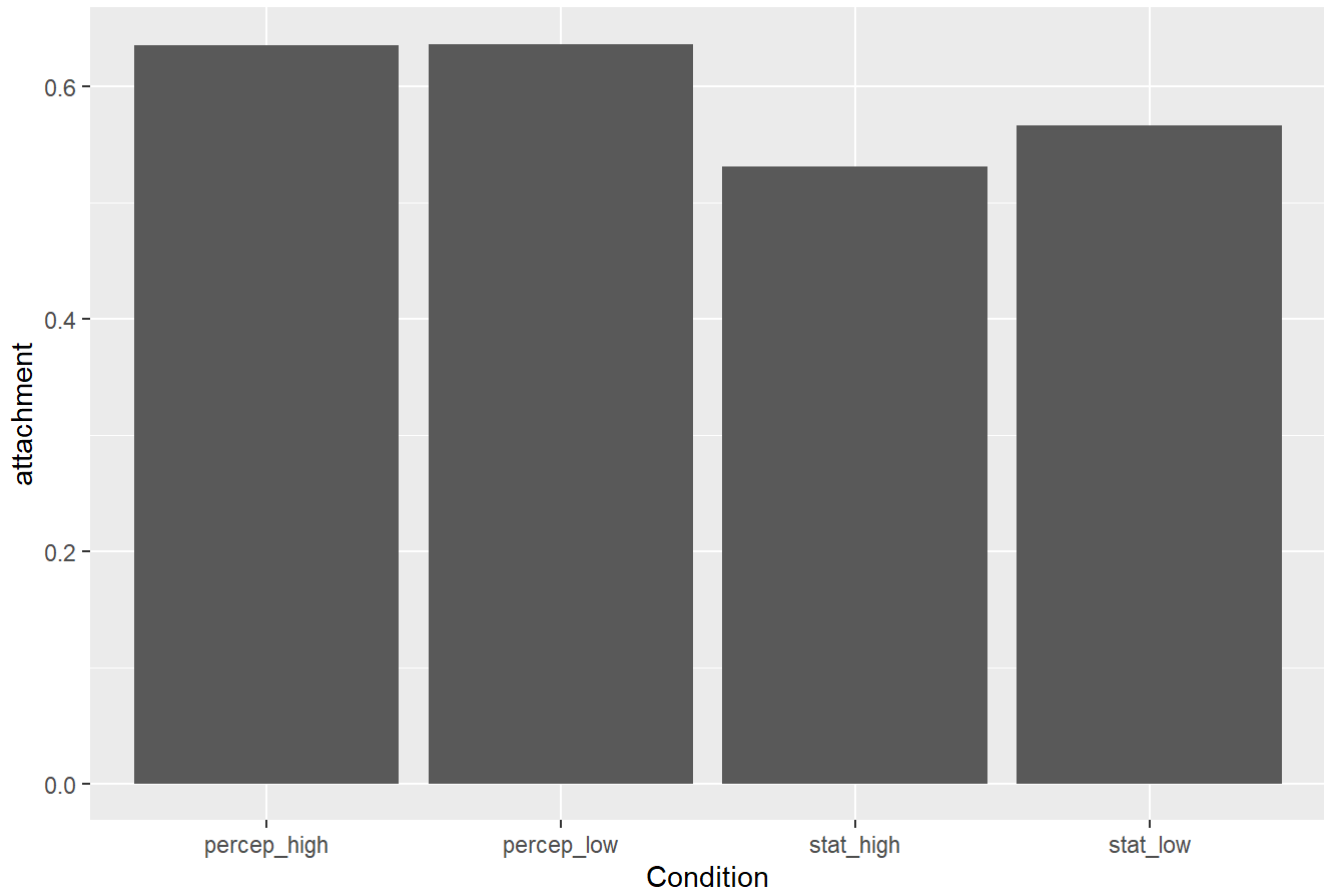
We can see that the condition percep returns more high-attachments than the stat condition.

I now plot the same graph (attachment for each condition) but with only correct answers to equation.

```
data.onlygood <- filter(data, data$cor.equation == 1)
ggplot(data=data.onlygood) +
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +
  ggtitle("Overall with only correct answers")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

Overall with only correct answers



The graph with and without wrong answers to equation respects the same pattern except for the stat condition: correct answers lead a slightly more high-attachments for the low condition.

To continue, I will split the data in two. Participant that are efficient to resolve equations and the others. To determine good participants, I will select those who have below 50% of correct answers.

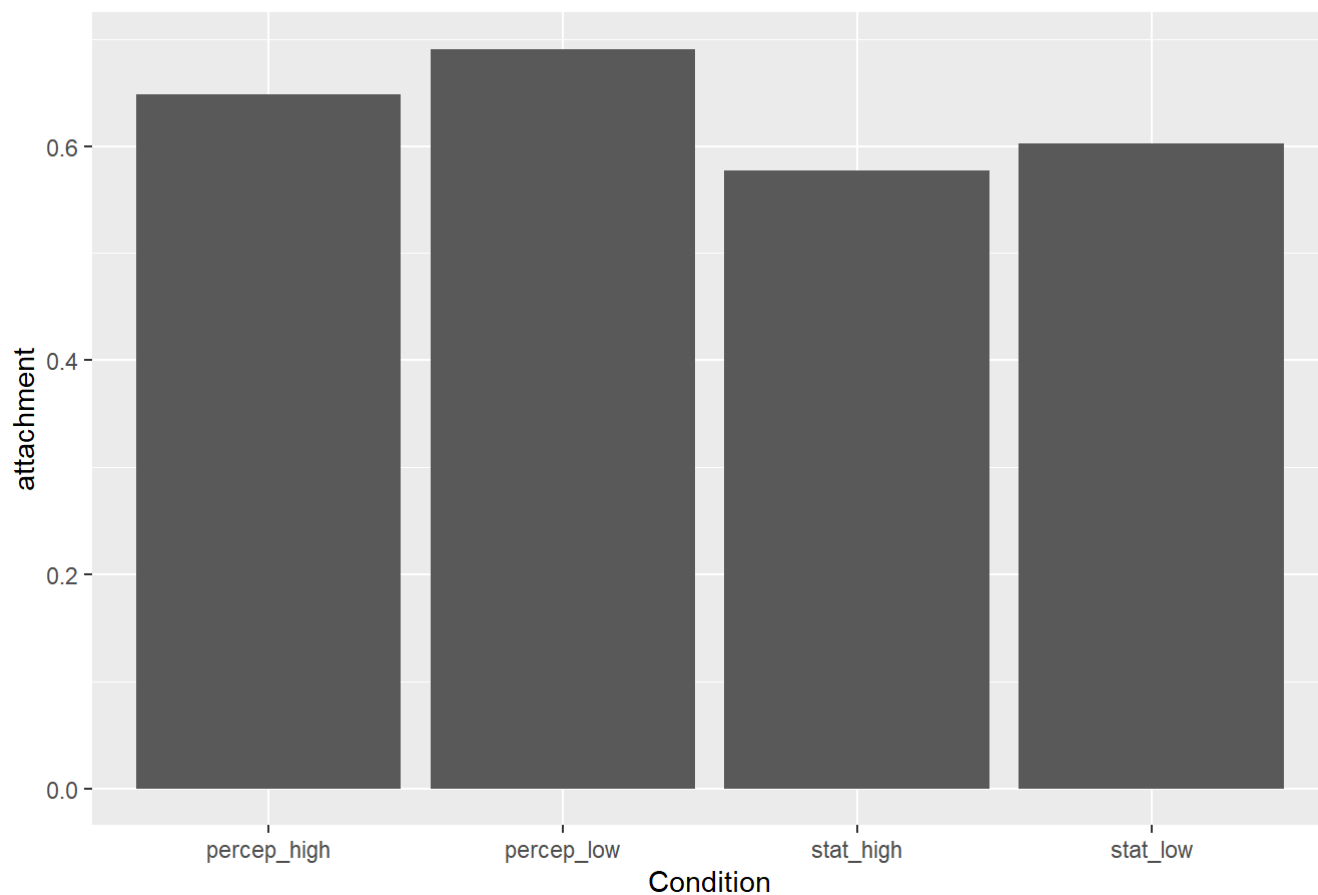
```
datamath <- filter(data, data$amount.cor.equation > 14)
datanomath <- filter(data, data$amount.cor.equation <= 14)
```

Then I plot those two groups.

```
ggplot(data=datamath) +
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +
  ggtitle("Efficient in math")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

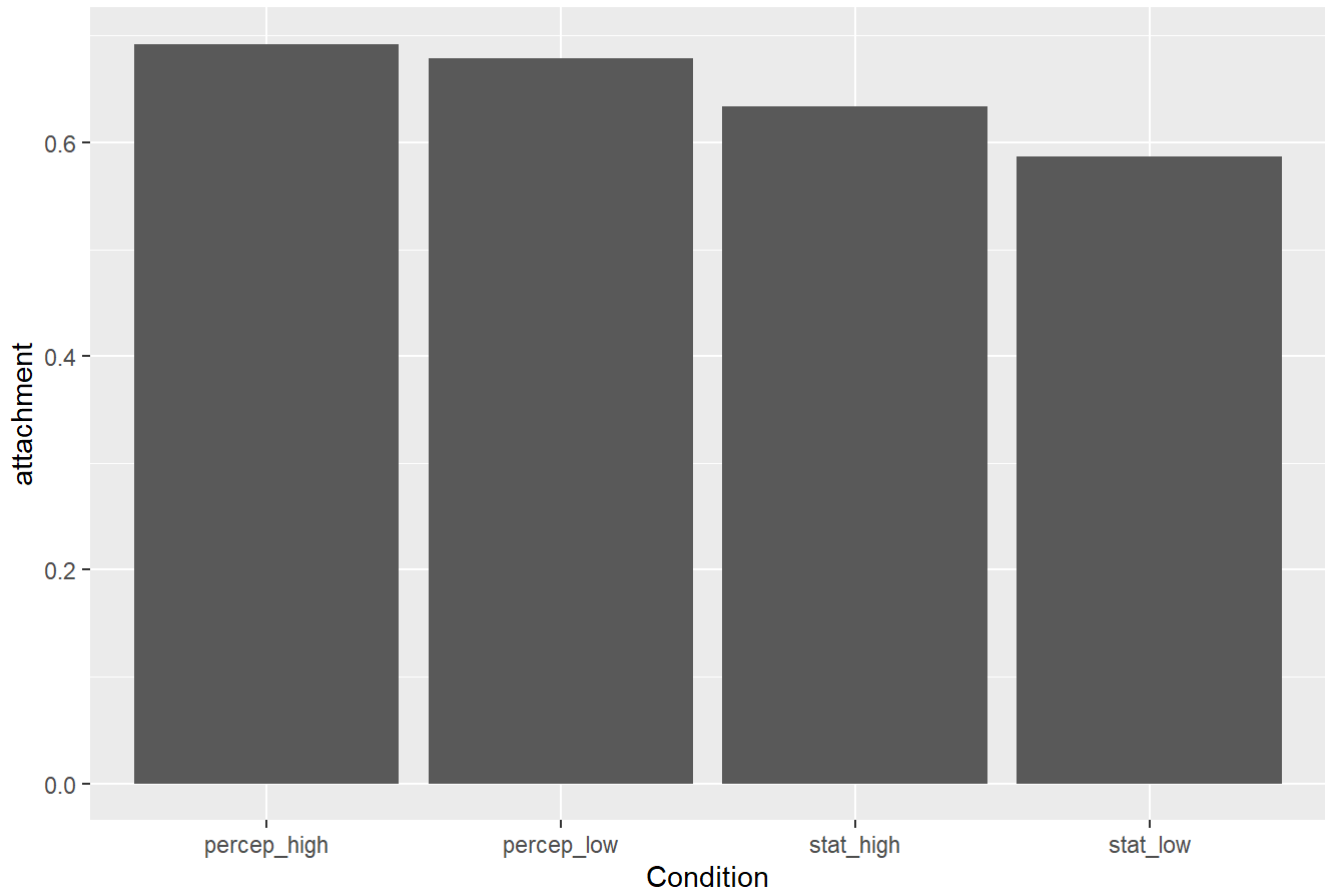
Efficient in math



```
ggplot(data=datanomath) +  
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +  
  ggtitle("Non-efficient in math")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

Non-efficient in math



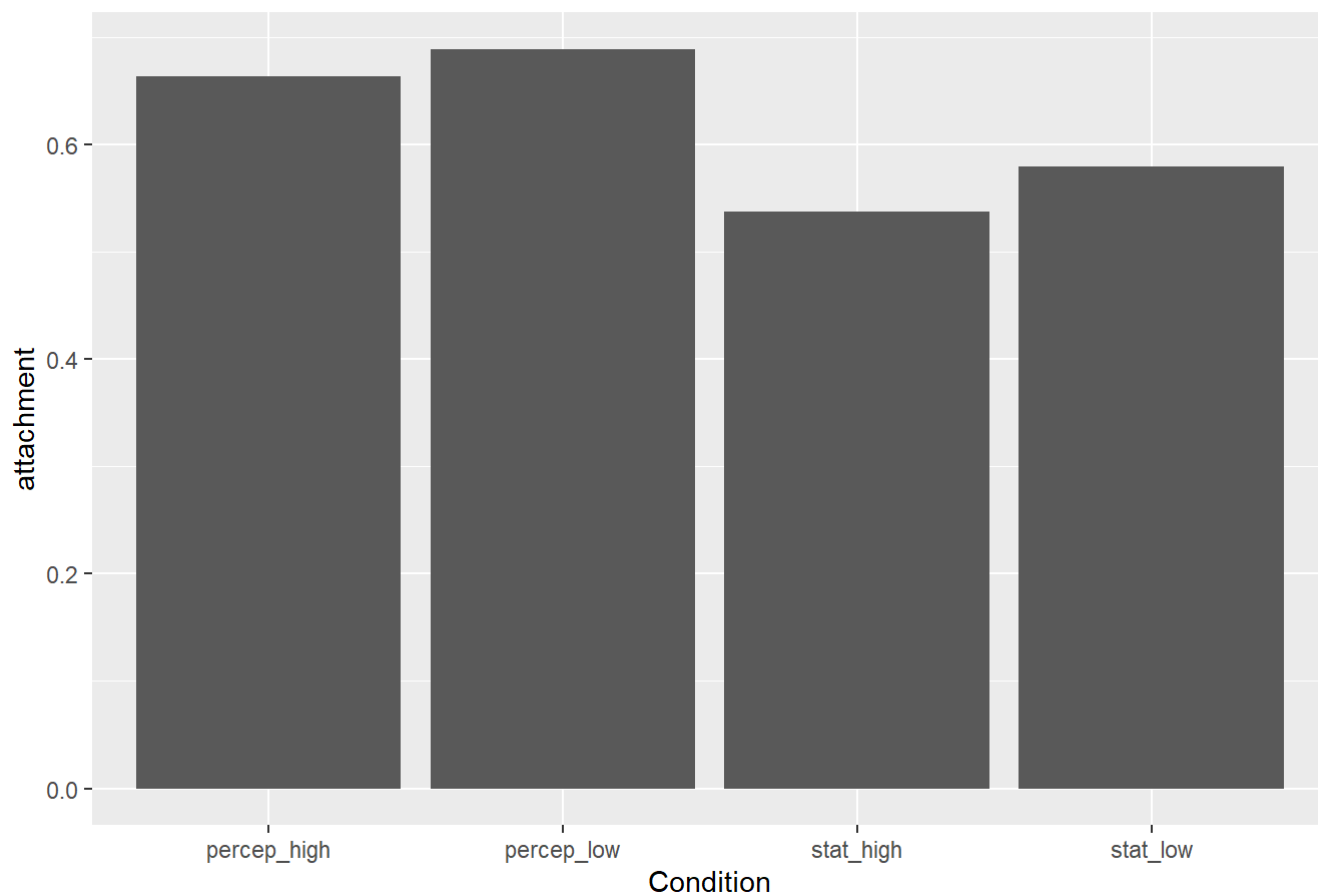
These are globally the same results, the only different trend is that the efficient group have slightly more high-attachments for the low condition whereas the non-efficient group have slightly more high-attachment for the high condition (which is contrary to the expected results, participants efficient in maths should be more subject to priming). Also the percep condition seems to return less high-attachments.

Next, I will select only correct answers to equation to report high-attachment (for both groups), and directly plot them.

```
datamath.onlygood <- filter(datamath, datamath$cor.equation == 1)
datanomath.onlygood <- filter(datanomath, datanomath$cor.equation == 1)
ggplot(data=datamath.onlygood) +
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +
  ggtitle("Efficient in math with only correct answers")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

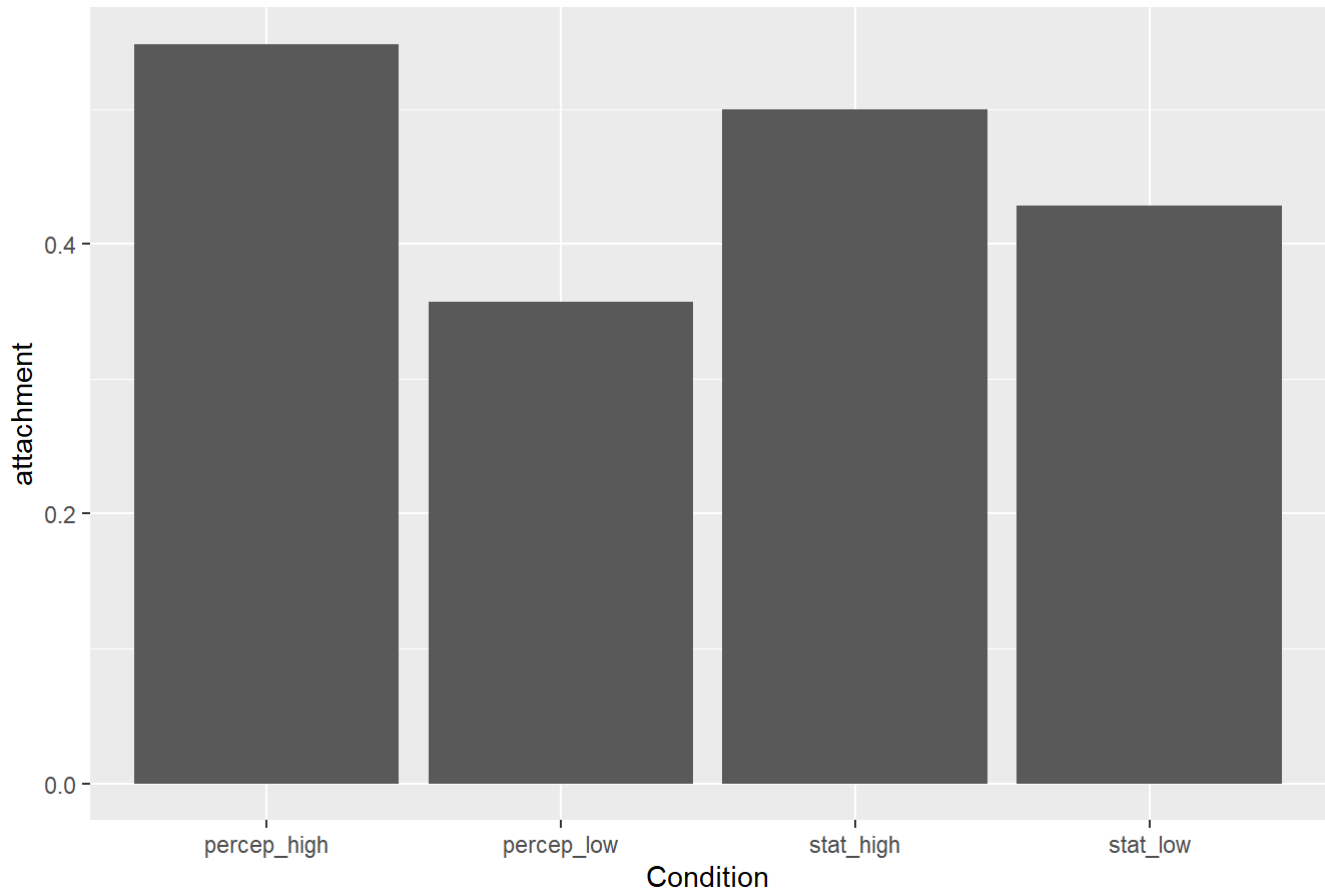
Efficient in math with only correct answers



```
ggplot(data=datanomath.onlygood) +  
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +  
  ggtitle("Non-efficient in math with only correct answers")
```

```
## No summary function supplied, defaulting to `mean_se()`
```


Non-efficient in math with only correct answers



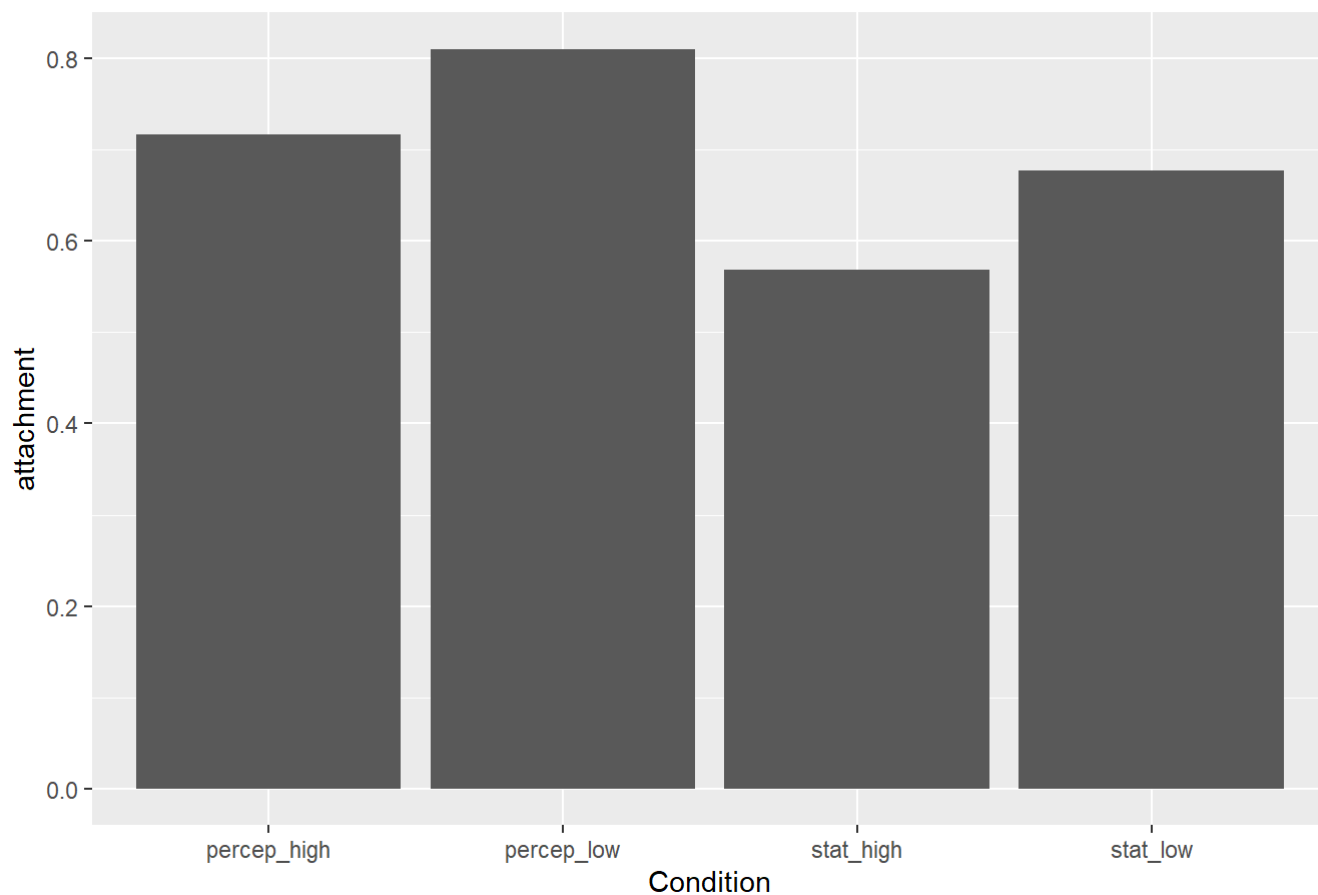
The graphs of the efficient group in math with and without good responses follow the same pattern. The non-efficient group shows a higher drop of high attachment in the low condition (as expected but too drastic for the percep condition which is not supposed to be subject to priming). However, this graph is plotted on a very limited set of trials (68). The percep condition now return more high attachment for the efficient group, but not for the non-efficient group.

Continuing with the efficient group with good answers, I plot the first and second half of the experiment.

```
datamath.onlygood.firsthalf <- filter(datamath.onlygood, datamath.onlygood$halfsession == 0)
datamath.onlygood.secondhalf <- filter(datamath.onlygood, datamath.onlygood$halfsession == 1)
ggplot(data=datamath.onlygood.firsthalf) +
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +
  ggtitle("Efficient in math with only correct answers, first half")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

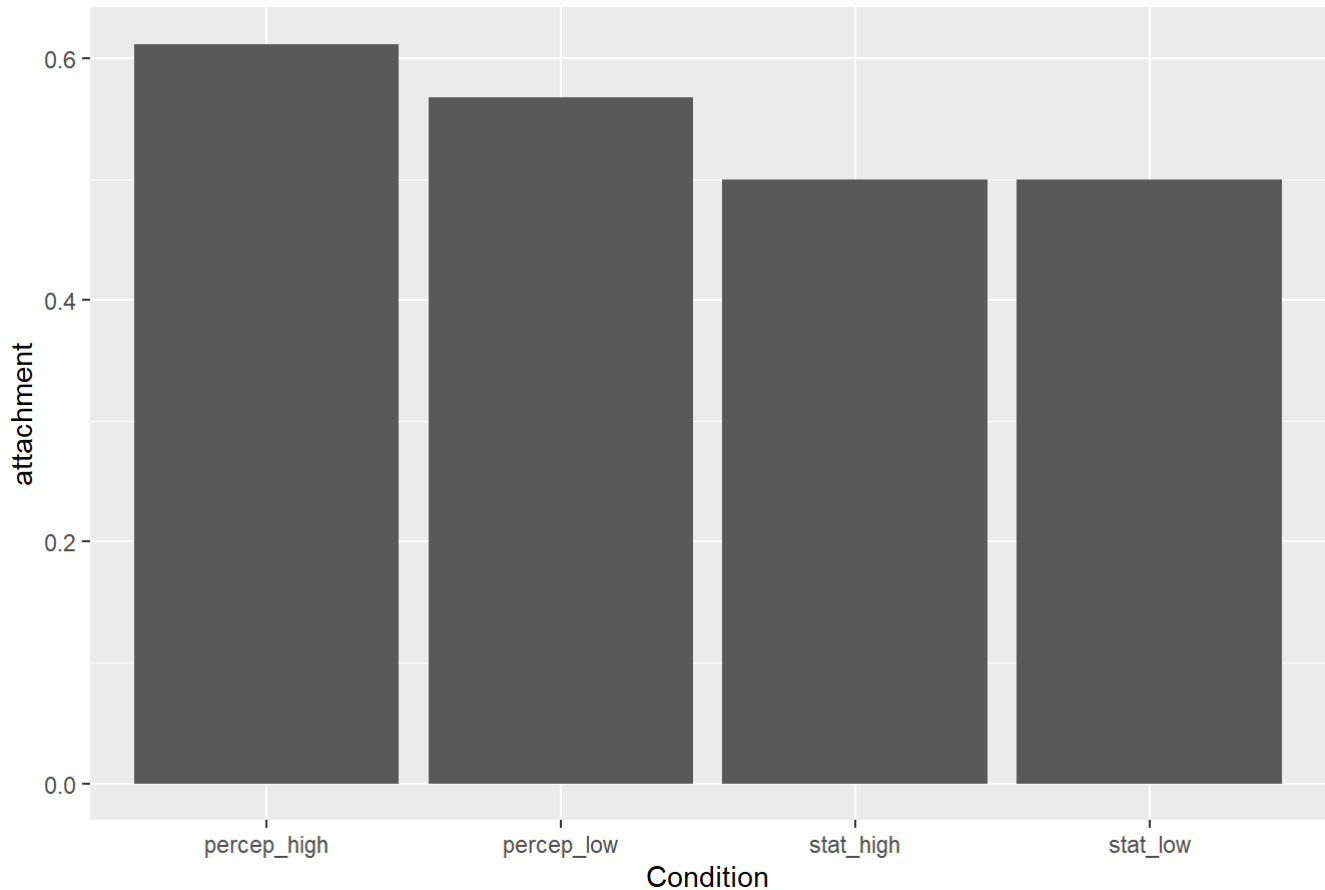
Efficient in math with only correct answers, first half



```
ggplot(data=datamath.onlygood.secondhalf) +  
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +  
  ggtitle("Efficient in math with only correct answers, second half")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

Efficient in math with only correct answers, second half



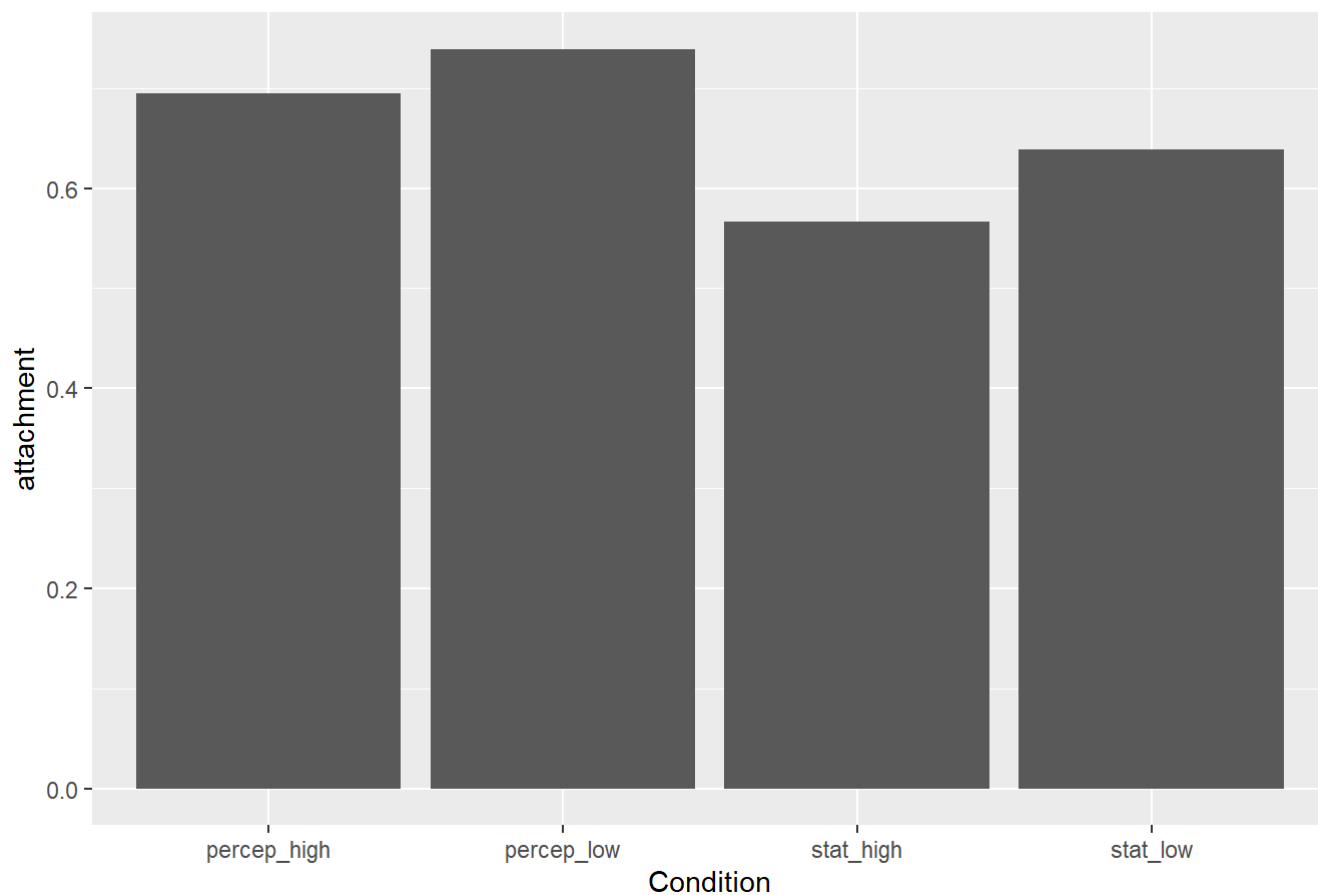
I can see that data in the first half yield more high-attachments in the low condition. Percep still return more high-attachments than stat. In the second half, no differences are shown between the high and low condition within the stat condition. Percep returned more high-attachments in the high condition than the low.

To see if we can observe such observation on all participant, I plot with all trials with good answers, first and second half

```
data.onlygood.firsthalf <- filter(data.onlygood, data.onlygood$halfsession == 0)
data.onlygood.secondhalf <- filter(data.onlygood, data.onlygood$halfsession == 1)
ggplot(data=data.onlygood.firsthalf) +
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +
  ggtitle("Overall with only correct answers, first half")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

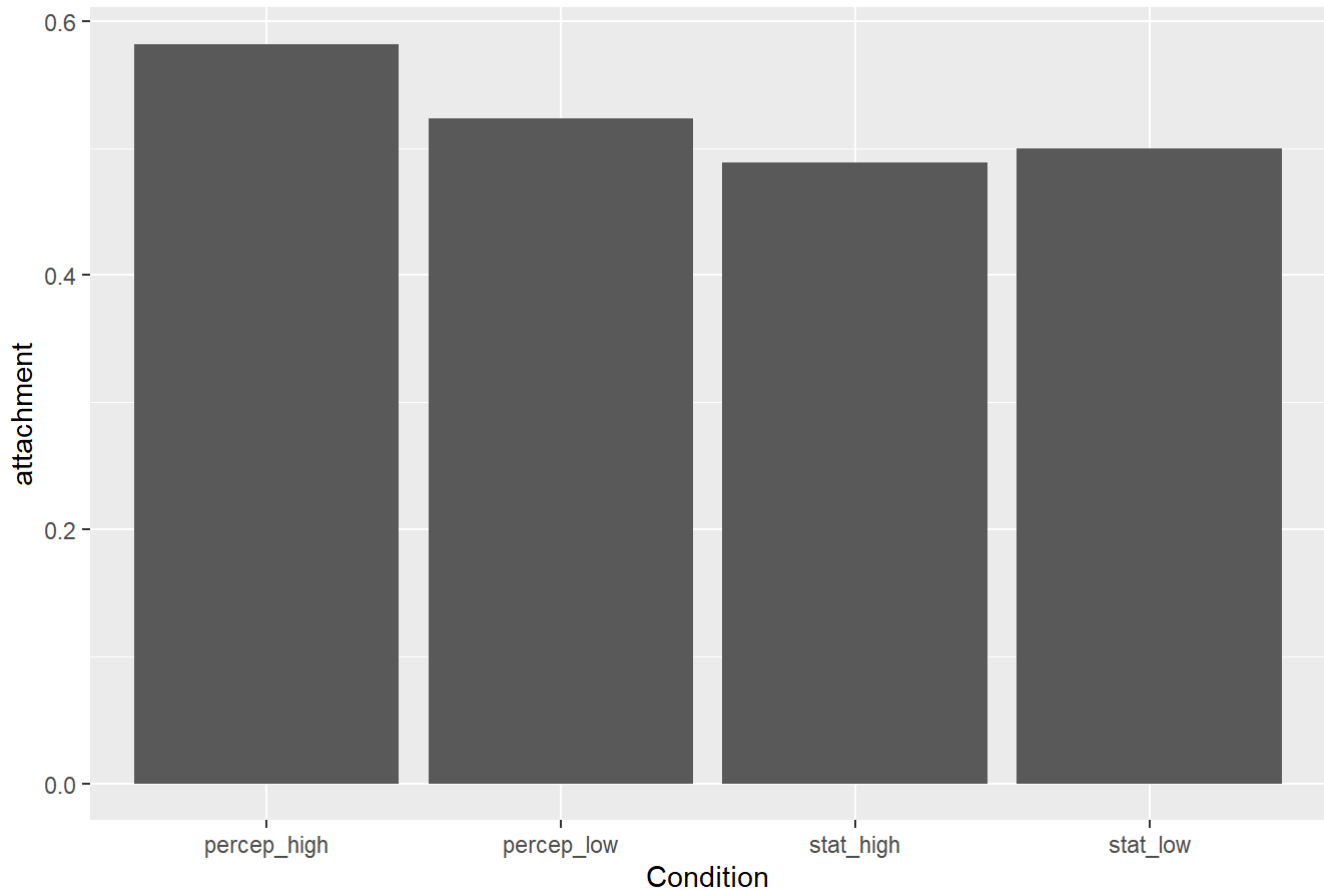
Overall with only correct answers, first half



```
ggplot(data=data.onlygood.secondhalf) +  
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +  
  ggtitle("Overall in math with only correct answers, second half")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

Overall in math with only correct answers, second half



We observe the same tendency, but once again, this may be due to the fact that only a small portion of correct answers within the non-efficient group are present.

On a new approach on this dataset, I will use reaction times to see if there are differences. I first determine the mean reaction time of both equation and completion (median could have been used as well)

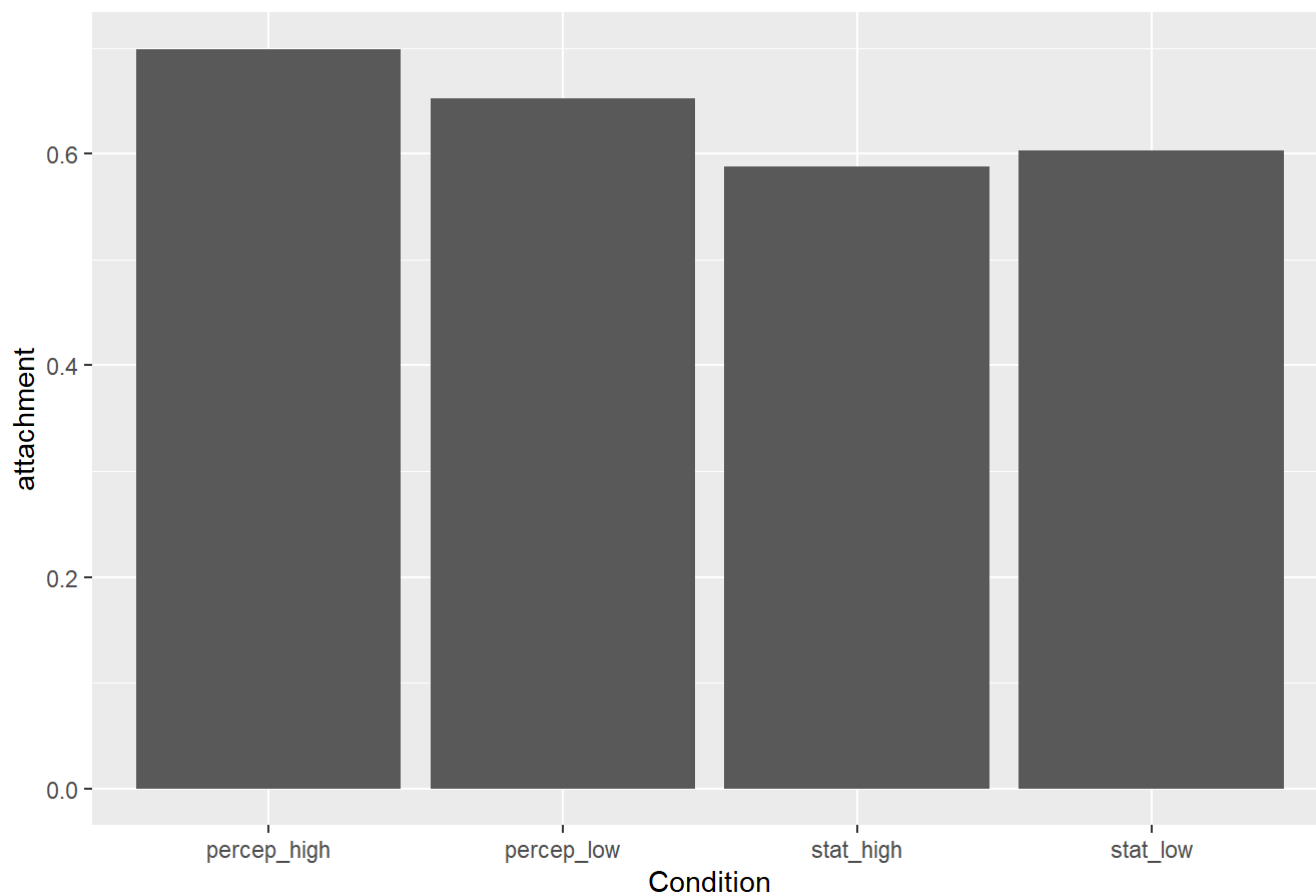
```
mean.comp <- mean(data$RT.completion)
mean.eq <- mean(data$RT.equation)
```

I First split groups between low and high equation RT and directly plot it

```
data.loweql <- filter(data, data$RT.equation < mean.eq)
data.higheql <- filter(data, data$RT.equation > mean.eq)
ggplot(data=data.loweql) +
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +
  ggtitle("Overall with low response time in equation")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

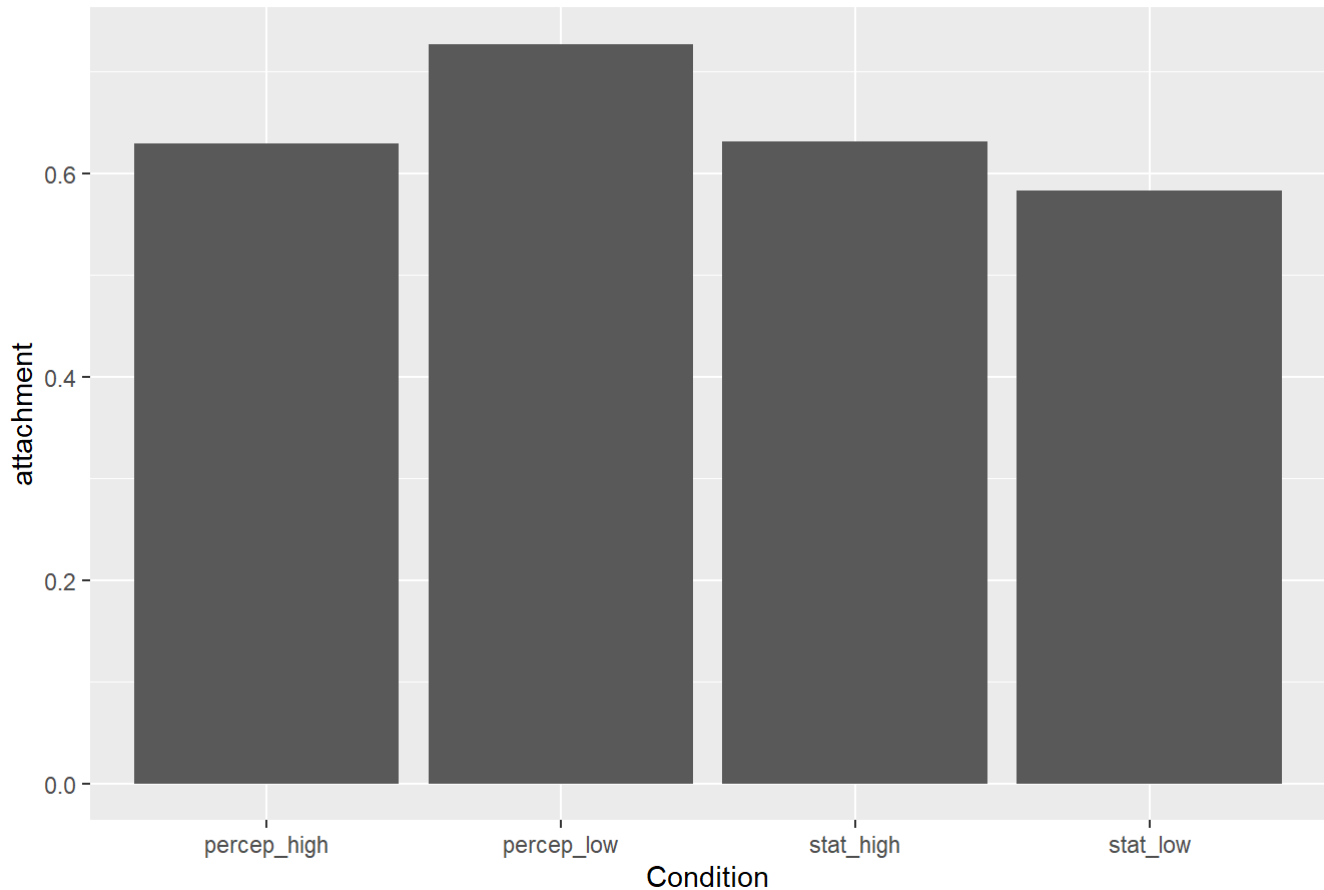
Overall with low response time in equation



```
ggplot(data=data.higheq) +  
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +  
  ggtitle("Overall with high response time in equation")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

Overall with high response time in equation



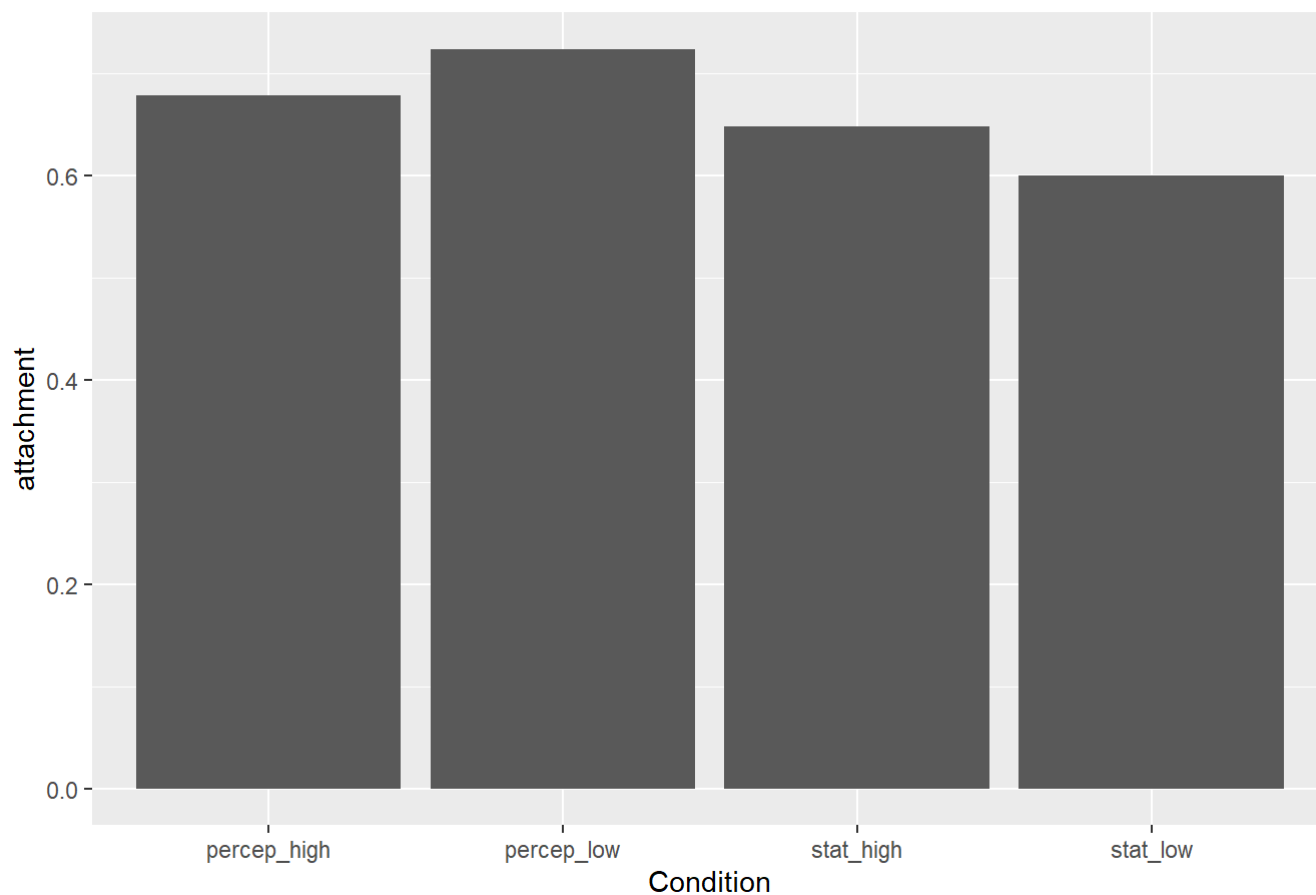
No major differences are observed for the stat condition. However, low response time in equation return more high-attachments in the high condition and high response time return more high-attachments in the low condition (equation for the percep condition should not impact attachment, but we see that response time for equation have a greater impact for the percep condition than the stat condition).

Second, I split groups between low and high completion RT and directly plot it

```
data.lowcomp <- filter(data, data$RT.completion < mean.comp)
data.highcomp <- filter(data, data$RT.completion > mean.comp)
ggplot(data=data.lowcomp) +
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +
  ggtitle("Overall with low response time in completion")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

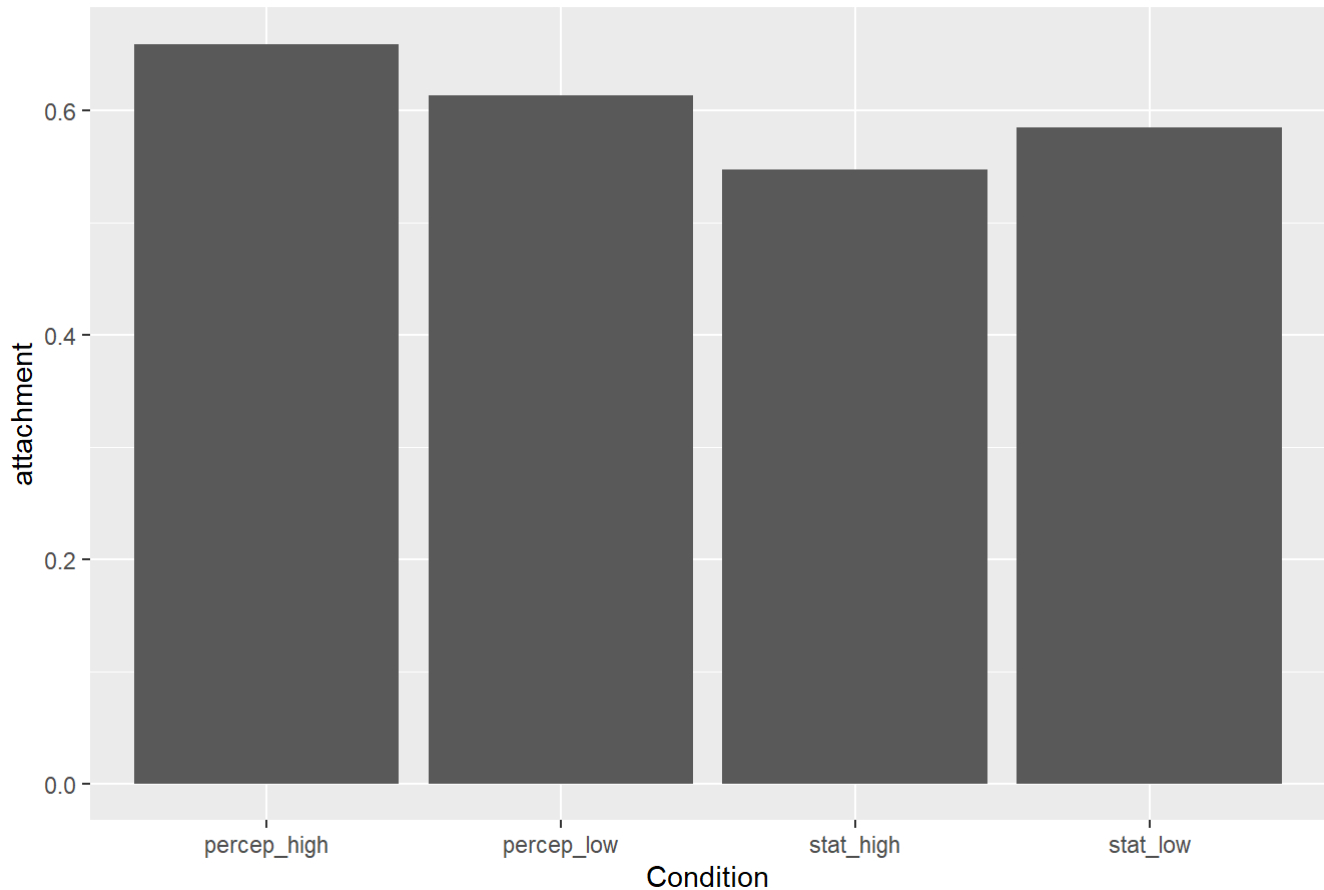
Overall with low response time in completion



```
ggplot(data=data.highcomp) +  
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +  
  ggtitle("Overall with high response time in completion")
```

```
## No summary function supplied, defaulting to `mean_se()`
```


Overall with high response time in completion



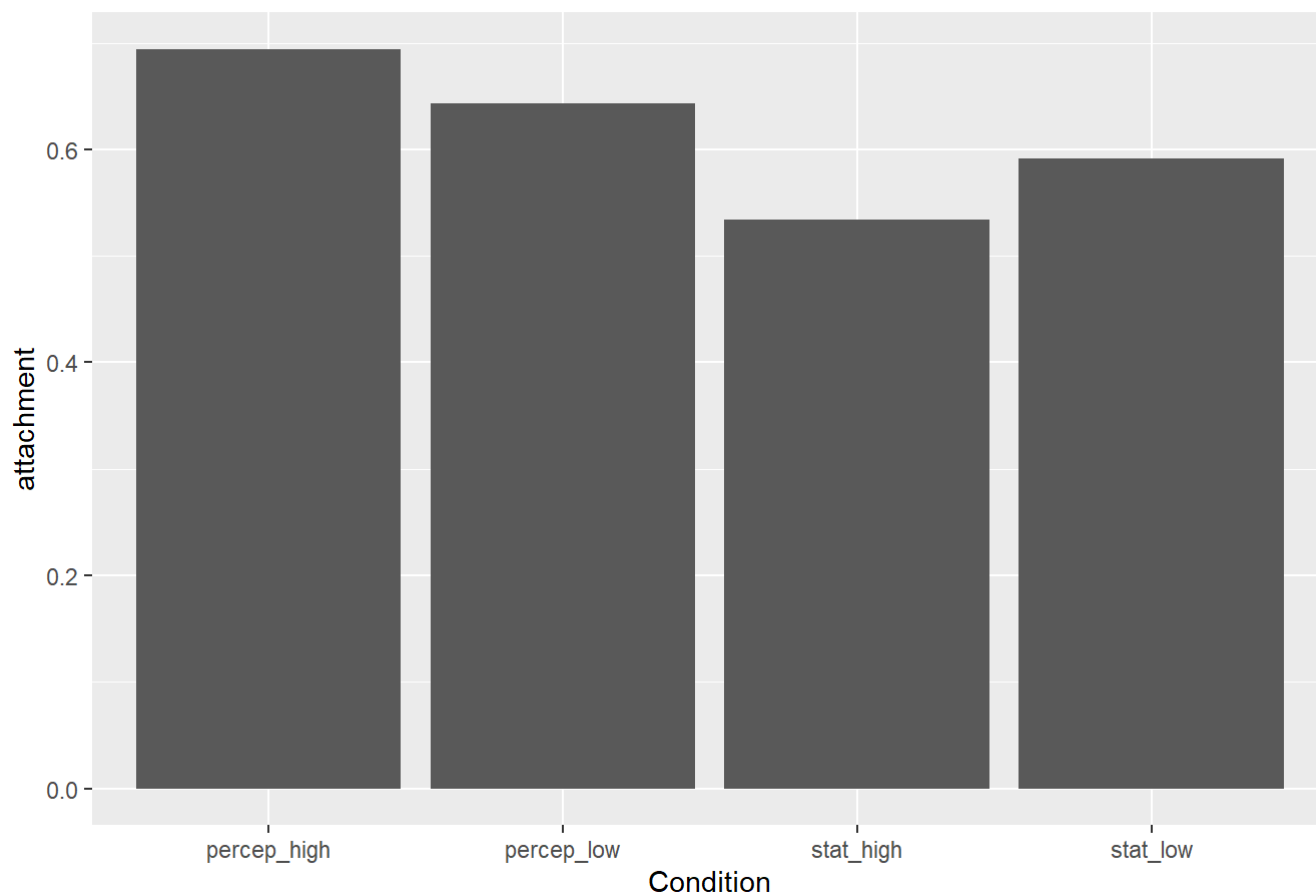
Here again, no major differences are observed but I will still talk about these little tendencies in accordance for my theory. High response time in completion yields more high-attachments in the high condition, the interpretation of a pseudo-relative seems to rise to over time. Low response time in completion yields more high-attachments in the high condition, the effect of priming seems to decay over time.

I now plot high-attachment in the efficient group with good responses for both completion and equation response time.

```
datamath.onlygood.lowe eq <- filter(datamath.onlygood, datamath.onlygood$RT.equation < mean.eq)
datamath.onlygood.higheq <- filter(datamath.onlygood, datamath.onlygood$RT.equation > mean.e
q)
datamath.onlygood.lowcomp <- filter(datamath.onlygood, datamath.onlygood$RT.completion < mea
n.comp)
datamath.onlygood.highcomp <- filter(datamath.onlygood, datamath.onlygood$RT.completion > mea
n.comp)
ggplot(data=datamath.onlygood.lowe eq) +
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +
  ggtitle("Efficient in math with only correct answers, low equation time")
```

```
## No summary function supplied, defaulting to `mean_se()
```

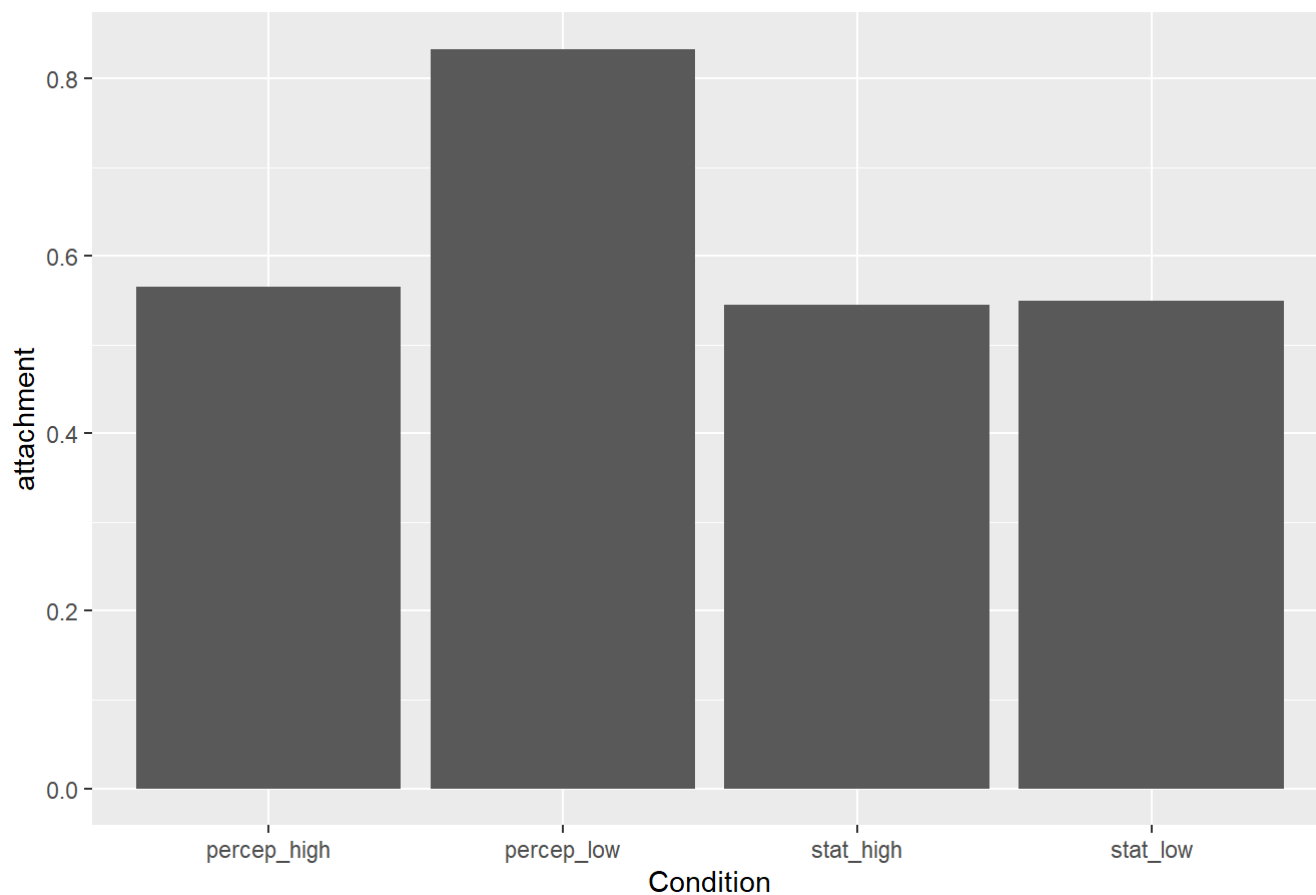
Efficient in math with only correct answers, low equation time



```
ggplot(data=datamath.onlygood.higheq) +  
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +  
  ggtitle("Efficient in math with only correct answers, high equation time")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

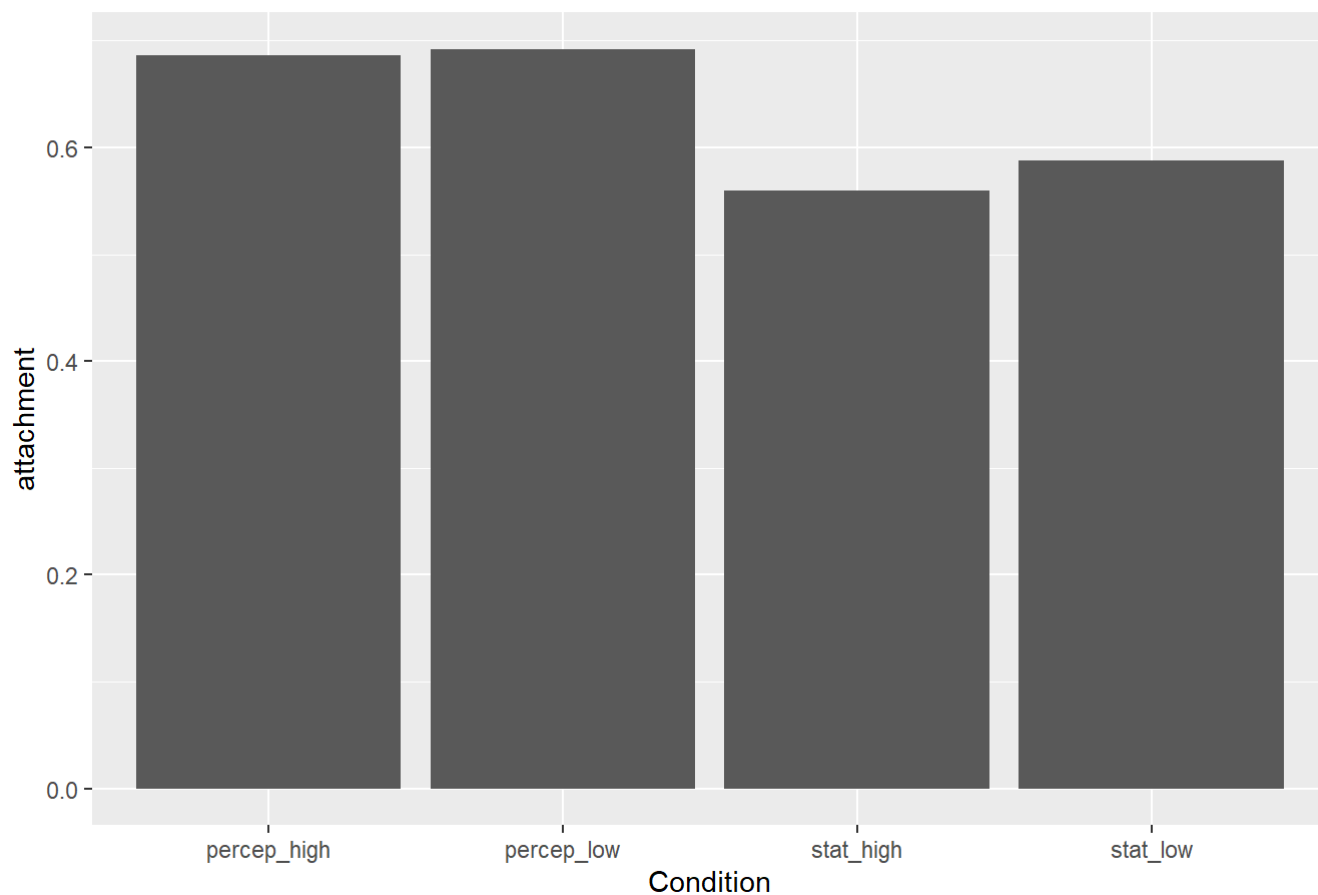
Efficient in math with only correct answers, high equation time



```
ggplot(data=datamath.onlygood.lowcomp) +  
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +  
  ggtitle("Efficient in math with only correct answers, low completion time")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

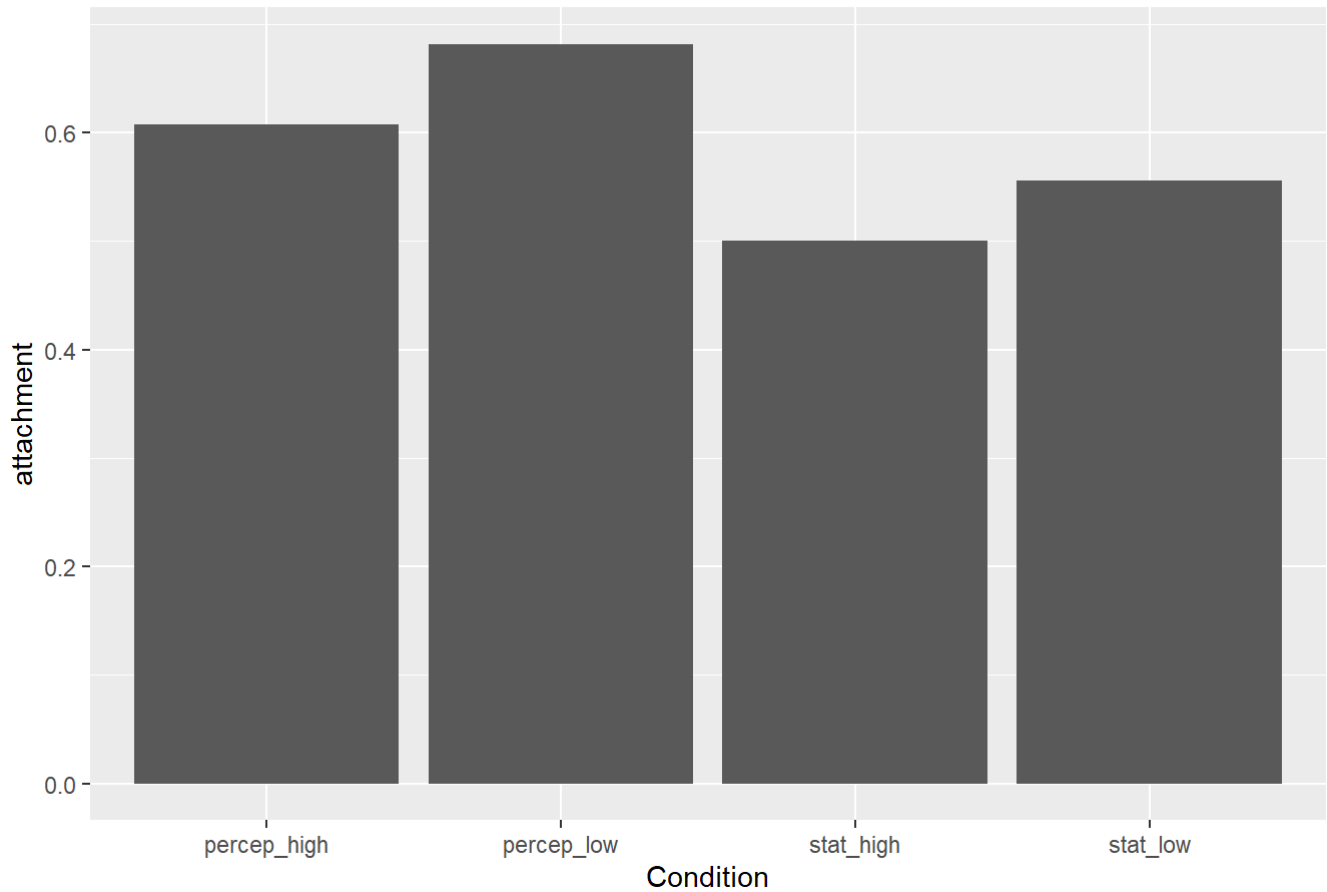
Efficient in math with only correct answers, low completion time



```
ggplot(data=datamath.onlygood.highcomp) +  
  geom_bar(mapping = aes(x = Condition, y = attachment), stat = "summary") +  
  ggtitle("Efficient in math with only correct answers, high completion time")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

Efficient in math with only correct answers, high completion time



We can see that low completion and low equation time give essentially the same pattern: the percep condition return more high-attachment (with slight difference for equation) and low condition returns more high-attachments within the stat condition. As for the high completion and equation time, the percep low condition returns a very high amount of high-attachments (this contradict what I plotted just before, the interpretation of the pseudo-relative does not rise over time).

\end{document}