

# Simple Barplot with error bars & data points using ggplot2

This script (Markdown file) was written to share what I find useful from my experience when plotting a simple (bar)chart on data with more than two factors, using ggplot. The plot includes visualisation of error bars and raw data points. Note: there would be better and more advanced scripts out there if that's what you are looking for.

In this example, I am using some data from two experiments included in this project:

<https://www.biorxiv.org/content/10.1101/473975v1> (<https://www.biorxiv.org/content/10.1101/473975v1>)

The plot to be created shows the effect of different factors on individual's acceptance/risk-taking rate

In the included .csv file here, each row contains data from a single trial. The column 'expt' specifies from which experiment the trial comes from, 'study1' or 'replicate' In column 'choice': '1' refers to 'accepting a gamble/risk'; '0' refers to 'rejecting a gamble/risk' The column 'raw\_rate' contains the participant's curiosity/desirability rating towards the stimulus presented in each trial (this won't be used in the current example) The column 'win\_prob' shows the probability of winning presented on a wheel of fortune in each trial

To start with, clear the current environment & then read into the .csv datafile

```
knitr::opts_chunk$set(error = TRUE)

#just having this habit to clear the global environment when starting some new computation
rm(list=ls())

#specify the path to your dataset (I am using the included datafile 'motivation_gamble_data.csv' as an example here)

##datafile <- "you file directory/motivation_gamble_data.csv"
datafile <- "D:/tryout/motivation_gamble_data.csv"

#to read a .csv file
df <- read.csv(datafile, header=TRUE, sep=",")
```

Data formatting to prepare for graph plotting In this example, we are plotting the acceptance rate (i.e. by averaging the responses in the 'choice' column) under a combination of different factors (1.'win\_prob'; 2.'category'; 3.'expt')

compile and create a dataframe that includes the mean, SD and SE of the acceptance rate

```

#first, to summarise the mean choices (i.e. acceptance rate) of different conditions (from the combination of the 3 factors) FOR EACH PARTICIPANT, using the aggregate function and FUN=mean (we will also use these data for plotting individual data-points in ggplot later)
accept_ppt_mean <- aggregate(choice ~ participant+category+win_prob+expt, data=df, FUN=mean)
#rename the last column which represents the mean choice (i.e. acceptance rate)
names(accept_ppt_mean)[length(accept_ppt_mean)]<- "mean_ppt_acceptrate"

#second, summarise the mean acceptance rate across all participants
accept_mean <- aggregate(mean_ppt_acceptrate ~ category+win_prob+expt, data=accept_ppt_mean, FUN=mean)
names(accept_mean)[length(accept_mean)]<- "mean_accept"

#compute the standard deviation of the participant's acceptance rate
accept_sd <- aggregate(mean_ppt_acceptrate ~ category+win_prob+expt, data=accept_ppt_mean, FUN=sd)
names(accept_sd)[length(accept_sd)]<- "sd_accept"

#combining the two dataframes with mean and sd acceptance rate data together
comb_df <- merge(accept_mean, accept_sd, by = c("category","win_prob","expt"))

#formulas to compute the standard errors of means (individual differences) to be used for plotting error bars
#may be different for different datasets depending on what your error bars represent
comb_df[,"se_accept"] <- NA

# 'study1' and 'replicate' under the 'expt' column come from two independent studies with different sample size and their SE should be computed separately
whichstudy1_idx <- which(comb_df$expt=="study1")
comb_df$se_accept[comb_df$expt=="study1"] <- comb_df[whichstudy1_idx,"sd_accept"]/(sqrt(length(unique(accept_ppt_mean$participant[accept_ppt_mean$expt=="study1"]))))

whichreplicate_idx <- which(comb_df$expt=="replicate")
comb_df$se_accept[comb_df$expt=="replicate"] <- comb_df[whichreplicate_idx,"sd_accept"]/(sqrt(length(unique(accept_ppt_mean$participant[accept_ppt_mean$expt=="replicate"]))))

```

For bar graph plotting, we need to install and use the `ggplots2` package. Specifically, we use 'geom\_bar' function in 'ggplot' to produce a 1d bar chart

```

#This line of code installs the ggplot2 package if you do not have it installed - if you do, it simply loads the package
if(!require(ggplot2))install.packages("ggplot2")

```

```

## Loading required package: ggplot2

```

```
library(ggplot2)
```

```
#graph plotting using ggplot function
```

```
#scale_fill_manual is used here to define colours of different bars representing different conditions
```

```
#facet_grid is used here to create multiple panels split by discrete variables (i.e. curiosity, food categories)
```

```
#ylim: set the limits of y-axis
```

```
new_plot <- ggplot(data= comb_df, aes(x=win_prob, y=mean_accept)) + facet_grid(.~category)+  
  geom_bar(aes(fill=expt), stat="identity", position=position_dodge()) +  
  scale_fill_manual(values = c("study1" = "orange", "replicate" = "steelblue")) +  
  facet_grid(.~category)+  
  ylim(0,1) + scale_x_discrete("Win Probability")
```

```
#adding onto the existing new_plot, these lines create the error bars using geom_errorbar function
```

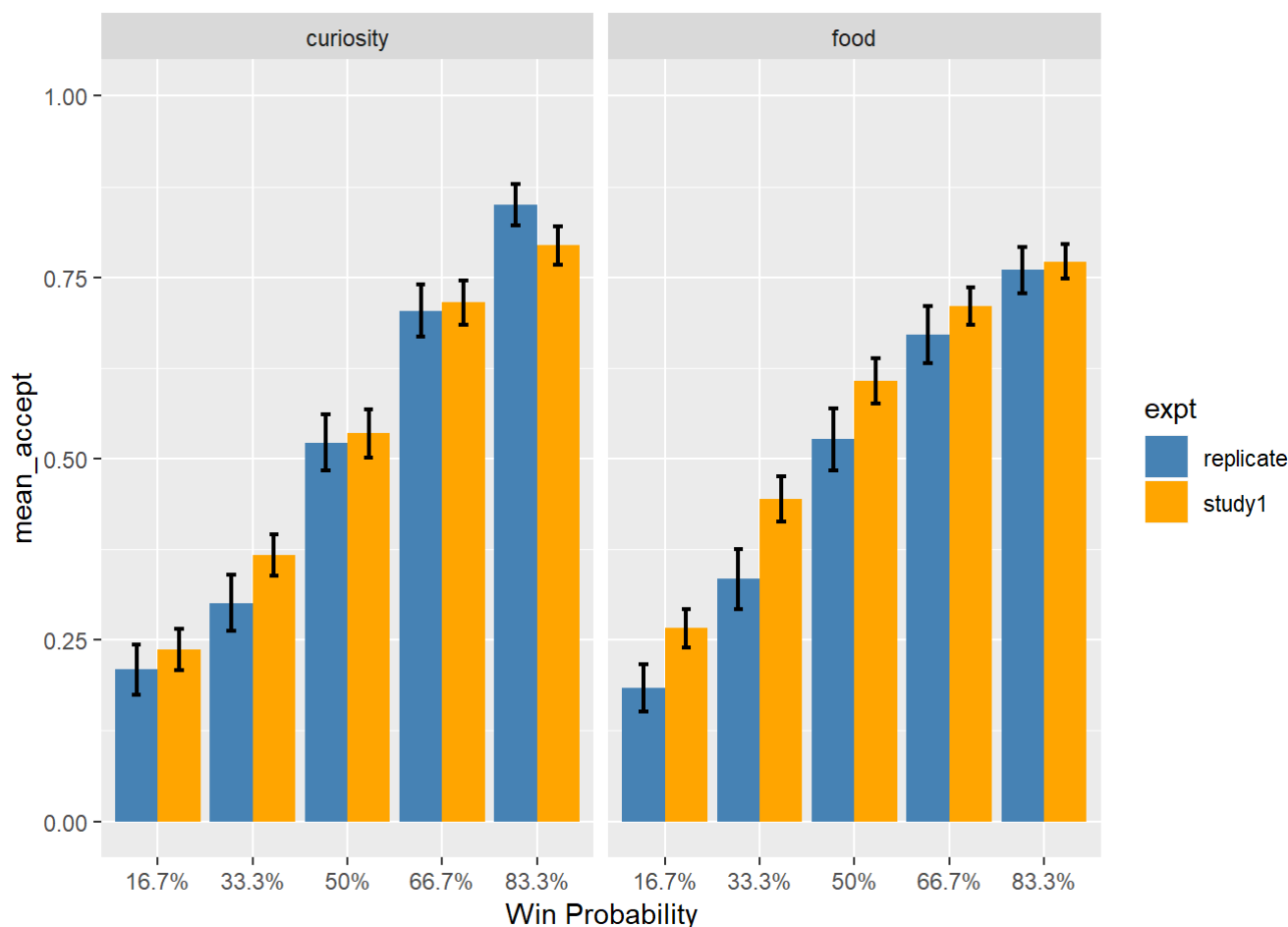
```
# >>specifying some parameters of the error bars
```

```
limits <- aes(x=win_prob, ymax=mean_accept+se_accept, ymin=mean_accept-se_accept, fill=expt)
```

```
new2_plot <- new_plot+ geom_errorbar(limits, position=position_dodge(width=0.9), width=0.2, size=0.75, color='black')
```

```
## Warning: Ignoring unknown aesthetics: fill
```

```
new2_plot
```



To also add data-points on the graph, `geom_point` function can be used

```
knitr::opts_chunk$set(warning = FALSE)
```

```
#play around with these parameters yourself to achieve the best jittering of your datapoints,  
using position_jitterdodge function
```

```
pos_dodg <- position_jitterdodge(jitter.width = 0.3, jitter.height = 0.001, dodge.width = 0.8  
5)
```

```
#note: the individual datapoint from the 'accept_ppt_mean' dataframe is used here
```

```
new2_plot + geom_point(data=accept_ppt_mean, aes(x=win_prob, y=mean_ppt_acceptrate,color=exp  
t), shape=21, position = pos_dodg) + scale_color_manual(values=c("darkorange","navyblue"))
```

```
## Warning: Removed 183 rows containing missing values (geom_point).
```

