Week 7 & 8 - Workbook

November 27, 2024

1 Week 7 & 8: Putting it all together!

1.1 Clear the entire workspace

[1]:
```
rm(list=ls())
```

1.2 Load libraries

[3]:
```
ReqdLibs =␣
 ↪c("here","ggplot2","dplyr","tidyr","stringr","janitor","broom","emmeans")
invisible(lapply(ReqdLibs, library, character.only = TRUE))
```

1.3 Check and set data directory

Recap Week 1 learning We want to know where we are and where we need to get the data from.

[5]:
```
# here()
folder_path = getwd()
# folder_path

subfolder_path = paste0(folder_path,'/data/')
# subfolder_path
# dir(subfolder_path)

file_list = list.files(subfolder_path)

file_list
```

1. 'QA1_T1_Barefoot1.txt' 2. 'QA1_T1_Barefoot2.txt' 3. 'QA1_T1_Barefoot3.txt'
4. 'QA1_T1_Barefoot4.txt' 5. 'QA1_T2_Barefoot1.txt' 6. 'QA1_T2_Barefoot2.txt'
7. 'QA1_T2_Barefoot3.txt' 8. 'QA1_T2_Barefoot5.txt' 9. 'QA1_T3_Barefoot4.txt'
10. 'QA1_T3_Barefoot5.txt' 11. 'QA1_T3_Barefoot6.txt'

1.4 Import data

Recap Week 2 learning We need some place to put the data and we don't want to read each file one by one.

[6]:
```
file_list = list.files(subfolder_path)
file_list
```

1. 'QA1_T1_Barefoot1.txt' 2. 'QA1_T1_Barefoot2.txt' 3. 'QA1_T1_Barefoot3.txt'
4. 'QA1_T1_Barefoot4.txt' 5. 'QA1_T2_Barefoot1.txt' 6. 'QA1_T2_Barefoot2.txt'
7. 'QA1_T2_Barefoot3.txt' 8. 'QA1_T2_Barefoot5.txt' 9. 'QA1_T3_Barefoot4.txt'
10. 'QA1_T3_Barefoot5.txt' 11. 'QA1_T3_Barefoot6.txt'

[8]:
```
file_path = paste0(subfolder_path,file_list[1])
# file_path
temp0 = read.delim(file_path)
# temp0
head(temp0,5)

new_names = paste(temp0[1,],temp0[4,])

colnames(temp0) = new_names

temp1 = temp0[-c(1:4),]
head(temp1,5)
```

A data.frame: 5 × 79

	X <chr>	QA1_T1_Barefoot1.c3d <chr>	QA1_T1_Barefoot1.c3d.1 <chr>	QA1_T1_Baref <chr>
1		Left Ankle Angular Position	Left Ankle Angular Position	Left Ankle Angu
2		LINK_MODEL_BASED	LINK_MODEL_BASED	LINK_MODEL
3		ORIGINAL	ORIGINAL	ORIGINAL
4	ITEM	X	Y	Z
5	1	4.026378	1.210450	-6.210010

A data.frame: 5 × 79

	ITEM <chr>	Left Ankle Angular Position X <chr>	Left Ankle Angular Position Y <chr>	Left Ankle <chr>
5	1	4.026378	1.210450	-6.210010
6	2	4.381411	0.430334	-6.378968
7	3	4.942418	-0.132680	-6.622121
8	4	5.649685	-0.382537	-6.946462
9	5	6.428778	-0.339809	-7.384519

[70]:
```
file_list = list.files(subfolder_path)
file_list
```

1. 'QA1_T1_Barefoot1.txt' 2. 'QA1_T1_Barefoot2.txt' 3. 'QA1_T1_Barefoot3.txt'
4. 'QA1_T1_Barefoot4.txt' 5. 'QA1_T2_Barefoot1.txt' 6. 'QA1_T2_Barefoot2.txt'
7. 'QA1_T2_Barefoot3.txt' 8. 'QA1_T2_Barefoot5.txt' 9. 'QA1_T3_Barefoot4.txt'
10. 'QA1_T3_Barefoot5.txt' 11. 'QA1_T3_Barefoot6.txt'

[84]:
```
new_names = paste(temp0[1,],temp0[4,])
```

[59]:
```
data.all = data.frame(list())
```

[67]:
```
for (i in 1:length(file_list)) {

file_path = paste0(subfolder_path,file_list[i])


temp = read.delim(file_path)
colnames(temp) = paste(temp[1,],temp[4,])
temp = temp[-c(1:4),]

temp = clean_names(temp)
colnames(temp)[1] = "perc_gait"
temp[,-1] = apply(temp[,-1],2,as.double)

temp$fileName = substr(file_list[i],1,nchar(file_list[i])-4)

data.all = rbind(data.all, temp)
}

head(data.all,6)
```

A data.frame: 6 × 80

	perc_gait <chr>	left_ankle_angular_position_x <dbl>	left_ankle_angular_position_y <dbl>	left_ <dbl
5	1	4.026378	1.210450	-6.21
6	2	4.381411	0.430334	-6.37
7	3	4.942418	-0.132680	-6.62
8	4	5.649685	-0.382537	-6.94
9	5	6.428778	-0.339809	-7.38
10	6	7.232831	-0.097753	-7.97

1.5 Data wrangling

Recap Week 3 learning We need some identifiers for plotting. We also don't want to plot all the variables one by one.

[68]:
```
data.all %>%
separate(fileName,sep = "_", into = c("prefix","session","trial"), remove =␣
 ↪FALSE) %>%
select(!prefix) %>%
{.->>data.clean}

head(data.clean)
```

A data.frame: 6 × 82

	perc_gait <chr>	left_ankle_angular_position_x <dbl>	left_ankle_angular_position_y <dbl>	left_ <db
5	1	4.026378	1.210450	-6.21
6	2	4.381411	0.430334	-6.37
7	3	4.942418	-0.132680	-6.62
8	4	5.649685	-0.382537	-6.94
9	5	6.428778	-0.339809	-7.38
10	6	7.232831	-0.097753	-7.97

[78]:
```
data.clean %>%
pivot_longer(cols = where(is.numeric), names_to = "measure", values_to =␣
 ↪"value") %>%
{.->> data.clean.longPlot}

data.clean.longPlot$perc_gait = as.double(data.clean.longPlot$perc_gait)
head(data.clean.longPlot)
```

A tibble: 6 × 6

perc_gait <dbl>	fileName <chr>	session <chr>	trial <chr>	measure <chr>	value <dbl
1	QA1_T1_Barefoot1	T1	Barefoot1	left_ankle_angular_position_x	4.026
1	QA1_T1_Barefoot1	T1	Barefoot1	left_ankle_angular_position_y	1.210
1	QA1_T1_Barefoot1	T1	Barefoot1	left_ankle_angular_position_z	-6.210
1	QA1_T1_Barefoot1	T1	Barefoot1	left_foot_progression_x	4.727
1	QA1_T1_Barefoot1	T1	Barefoot1	left_foot_progression_y	1.860
1	QA1_T1_Barefoot1	T1	Barefoot1	left_foot_progression_z	-2.955

1.6 Visualize data

Recap Week 4 learning We need an initial version of a plot where we can see all our data as means and standard errors. We want its appearance to be large and clear enough to be visible to the (relatively old) human eye.

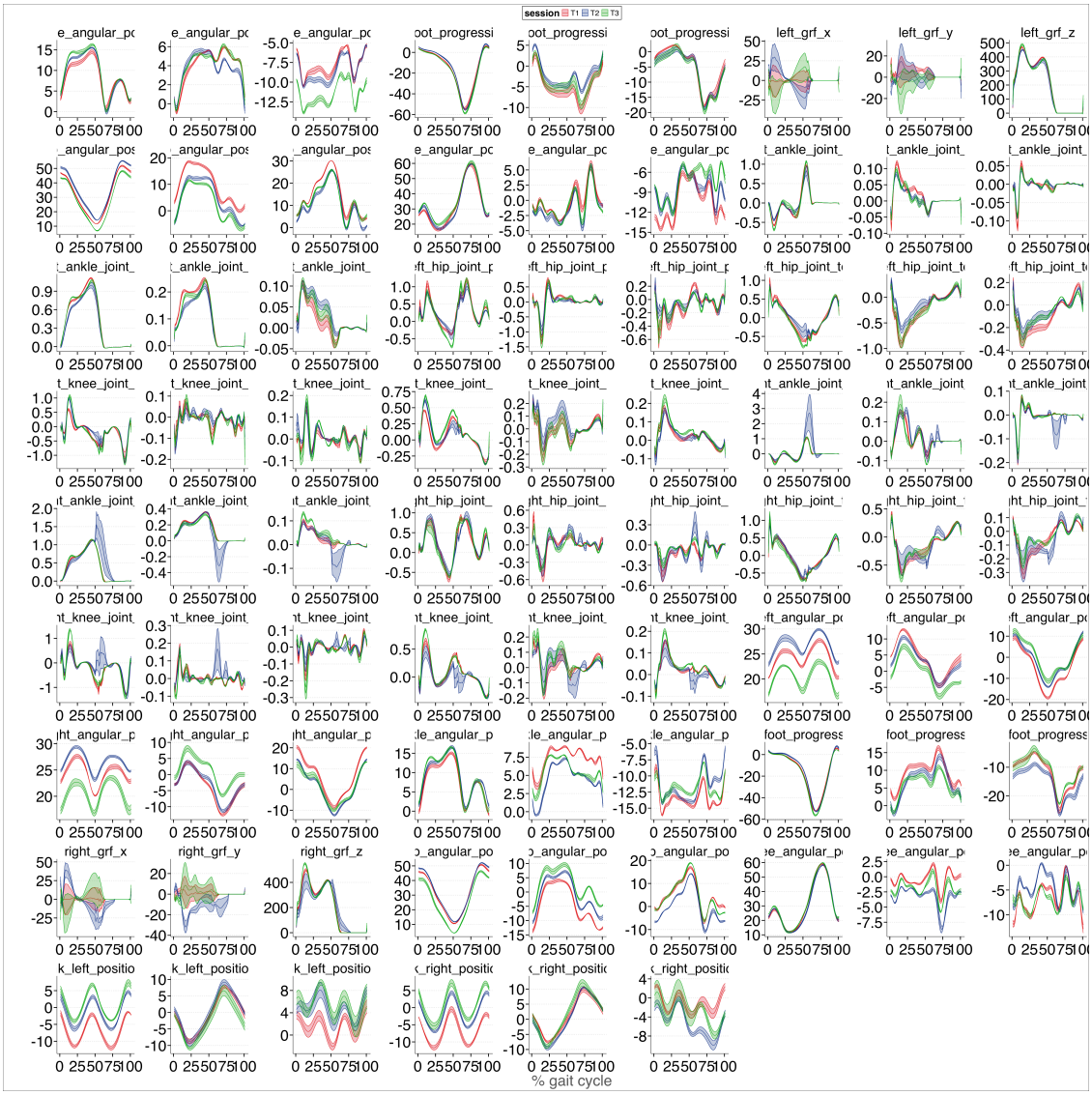
[80]:
```
thm = theme(plot.title = element_text(size = 40),
          legend.title = element_text(size = 25),
          legend.position = "top",
          legend.text = element_text(size = 20),
          strip.text = element_text(size = 35),
          axis.ticks.length = unit(0.3,"cm"),
          axis.line = element_line(colour = "black",linewidth = 1),
          axis.ticks = element_line(colour = "black",linewidth = 1),
          axis.text = element_text(colour = "black",size = 40),
          axis.text.x = element_text(lineheight = 1.1, margin = margin(t = 10)),
          axis.title.x = element_text(size=40, colour = "grey35", face =␣
 ↪"plain",
                                      lineheight = 1.1, margin = margin(r = 10)),
          axis.title.y = element_text(size=40, colour = "grey35", face =␣
 ↪"plain",
                                      lineheight = 1.1, margin = margin(r = 10)))
```

```
custom_colors <- c("#e41a1c", "#13388e", "#03ac13")
```

[81]:
```
library("ggthemes")
```

[83]:
```
options(repr.plot.width = 40, repr.plot.height = 40)
all_vars =
ggplot(data.clean.longPlot, aes(x = perc_gait,y = value,
                                group = session, col = session, fill =␣
 ↪session)) +
stat_summary(geom = "line", fun = mean, na.rm = TRUE) +
stat_summary(geom = "ribbon",fun.data = mean_se, na.rm = TRUE,alpha=0.3) +
  scale_color_manual(values = custom_colors) +
  scale_fill_manual(values = custom_colors) +
xlab("% gait cycle") + ylab("") +
facet_wrap(~measure, scales = "free")  +
theme_clean() + thm
all_vars

# ggsave(file='all_vars.svg', plot=all_vars, width=35, height=35)
```



1.7 Advanced data wrangling

Recap Week 4 learning What's the problem with the plot above?

[ ]:

1.8 What did you find from visual analysis of the data?

[ ]:

1.9 Statistics

Recap Week 5 learning We want to compare if some of these measures are significantly different between the 3 sessions.

[ ]:

2 Conclusion

What did you find?

[ ]: