Circle Plot: sleep deprivation

```
# Final plot here
```

Here I will further investigate the use of circle plot by focusing on a different aspect.

First, I combine the data probe and the sleep. The data used for this exploration has been downloaded from OSF (https://osf.io/xq6wr/).

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.2     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.2     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
# loader script
list.files("../data", pattern=".rdata", full.names = T,ignore.case = T) -> files
for(x in files){load(x)}
```

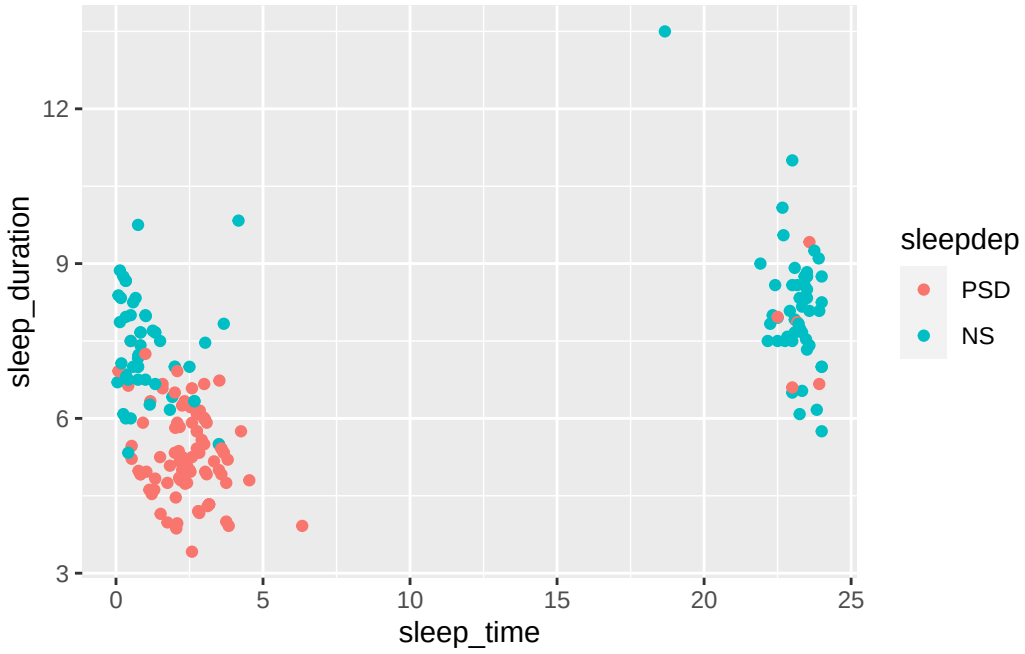
Before we start, some minor changes to the data frame will be done to make the plots look better and make them easier to work with.

```
sleep_diary |>
  mutate(sleepdep = factor(if_else(pre_control==1, "NS",
                           ifelse(pre_sleepdep==1, "PSD", NA)), levels = c("PSD","NS")),
         subj = as.numeric(subj)
         ) |>
  filter(!(is.na(sleepdep))) -> sleep
```

According to the draft idea, I believe it is best to plot the data according to sleep time and sleep duration. More specifically, the x-variable containing the time that participants fell asleep, and the y-variable relating to the amount of sleep participant received. This will be split into their respective conditions and in this way, we should see that participants go to sleep later, and sleep less.

```
sleep |>
  ggplot(aes(x=sleep_time, y=sleep_duration, col=sleepdep))+
  geom_point()
```

```
Warning: Removed 8 rows containing missing values (`geom_point()`).
```



As expected, participants did go to sleep later during the PSD (SD) condition and slept fewer hours as compared to the NS (control). However, a problem with this plot is that it visualizes all the (three) data points pertaining to participants sleep in each condition. To more effectively visualize with a circle plot, I should use a summary statistic of each the sleep.

To illustrate my point, we can visualize sleep duration over each subject, split by condition.

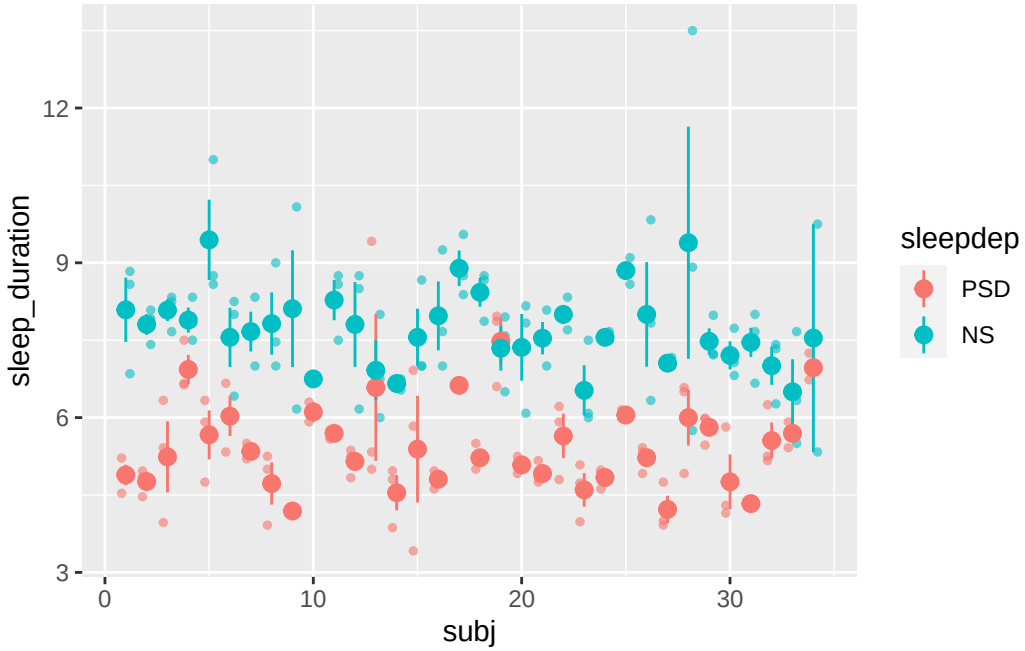
```
sleep |>
  ggplot(aes(x=subj, y=sleep_duration, col=sleepdep))+
  geom_point(position=position_dodge(.8), size=1, alpha=.6)+
  stat_summary()
```

```
Warning: Removed 8 rows containing non-finite values (`stat_summary()`).
```

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

```
Warning: Removed 8 rows containing missing values (`geom_point()`).
```

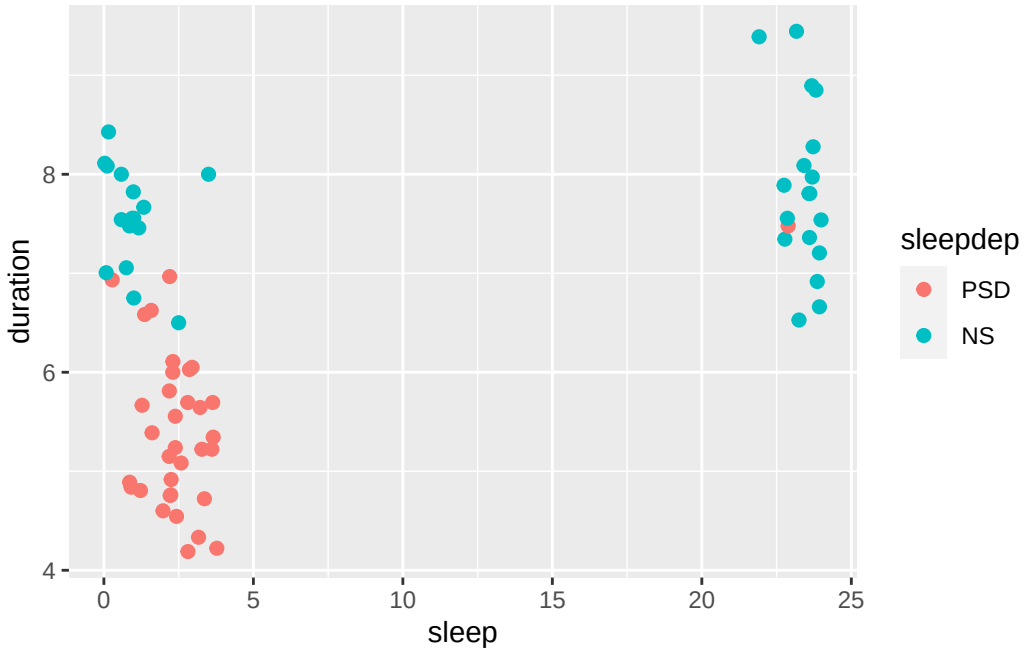
```
Warning: Removed 1 rows containing missing values (`geom_segment()`).
```



This plot produce a summary statistic (mean_se) over each subject for each condition. The result is more of what I am looking for. However, this plot is missing the relation to sleep time, which has to be represented by one of the axis. There is a problem related to this, however, in that each sleep duration point has a related sleep time point. This means that there is an association between all the data points for our plot, and grouping/summarizing within the ggplot is hard (if not impossible). One work around is to transform the data before we plug it into ggplot.

```
sleep |>
  group_by(subj) |>
  summarize(
    # Mean sleep duration over participant by condition
    NS_duration = mean(sleep_duration[sleepdep=="NS"], na.rm=T),
    PSD_duration = mean(sleep_duration[sleepdep=="PSD"], na.rm=T),
    # mean sleep time over participant by condition
    NS_sleep = mean(sleep_time_cum[sleepdep=="NS"], na.rm=T),
    NS_sleep = ifelse(NS_sleep>=24, NS_sleep-24, NS_sleep),
      #' Fix to standard clock time
    PSD_sleep = mean(sleep_time_cum[sleepdep=="PSD"], na.rm=T),
    PSD_sleep = ifelse(PSD_sleep>=24, PSD_sleep-24, PSD_sleep),
    # mean wake time over participant by condition
    NS_wake = mean(last_awaking_fix[sleepdep=="NS"], na.rm=T),
```

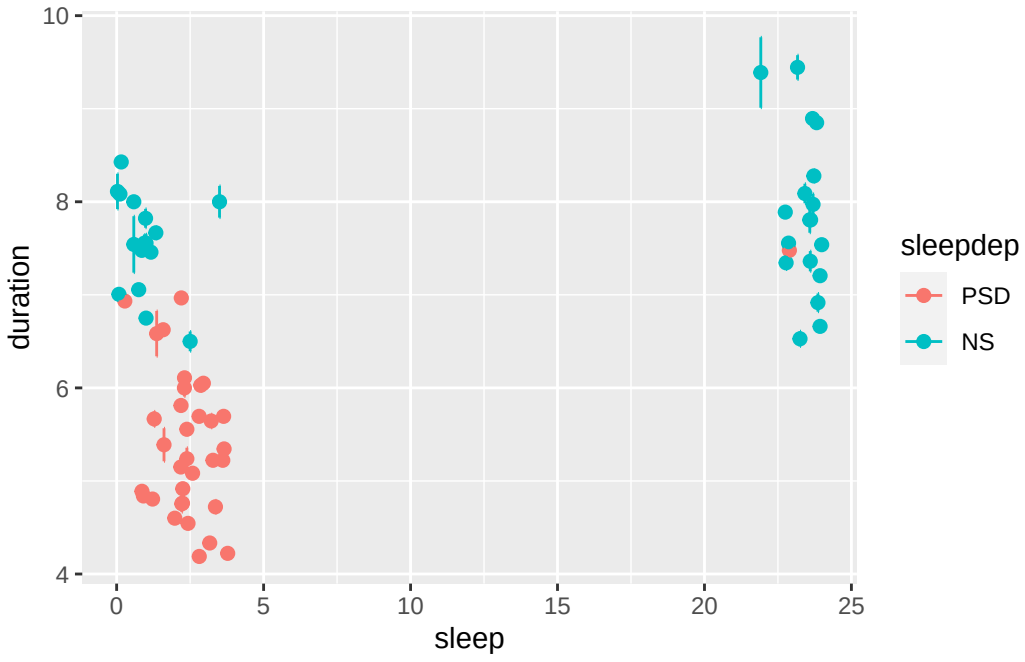
```
    PSD_wake = mean(last_awaking_fix[sleepdep=="PSD"], na.rm=T),
    ) |>
  pivot_longer(c(starts_with("NS"), starts_with("PSD")),
               names_pattern = "(.*)_(.*)",
               names_to=c("sleepdep", ".value")) |>
  mutate(sleepdep=factor(sleepdep, levels=c("PSD", "NS"))) |>
  ggplot(aes(x=sleep, y=duration, fill=sleepdep, col=sleepdep))+
  geom_point(size=2)
```



This result in a plot that seem to contain more of what I am looking for. Another thing that could be added to the plot is some uncertainty relating to the data. For instance, by using standard error, standard deviation or confidence intervals. For now I will stick with standard error, but this can easily be changed later.

```
sleep |>
  group_by(sleepdep) |>
  mutate(sd_length = length(sleepdep)) |> ungroup() |>
  group_by(subj, sleepdep) |>
  reframe(
    # Mean sleep duration
    duration = mean(sleep_duration, na.rm=T),
```

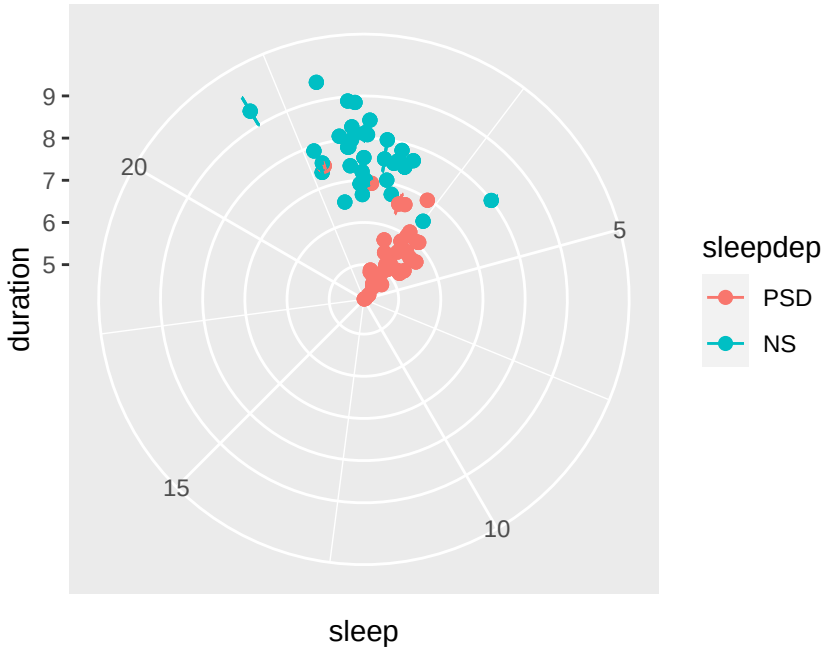
```
    duration_sd = sd(sleep_duration, na.rm=T),
    duration_se = sd(sleep_duration, na.rm=T) / sqrt(sd_length),
    # mean sleep time over participant by condition
    sleep = mean(sleep_time_cum, na.rm=T),
    sleep = ifelse(sleep>=24, sleep-24, sleep),
    sleep_sd = sd(sleep_time_cum, na.rm=T),
    sleep_se = sd(sleep_time_cum, na.rm=T) / sqrt(sd_length),
    # mean wake time over participant by condition
    waking = mean(last_awaking_fix, na.rm=T),
    waking_sd = sd(last_awaking_fix, na.rm=T),
    waking_se = sd(last_awaking_fix, na.rm=T) / sqrt(sd_length),) |>
  unique() -> s_sleep

s_sleep |>
  ggplot(aes(x=sleep, y=duration, col=sleepdep))+
  geom_point(size=2)+
  geom_errorbar(aes(ymin=duration-duration_se, ymax=duration+duration_se))+
  geom_point(aes(group=subj))
```



Here I improved upon the data transformation and included a calculation of the standard error and standard deviation. This will be used to indicate some uncertainty in our estimate or indicate the variation of the data. As noted previously, circle plots are more apt at visualizing

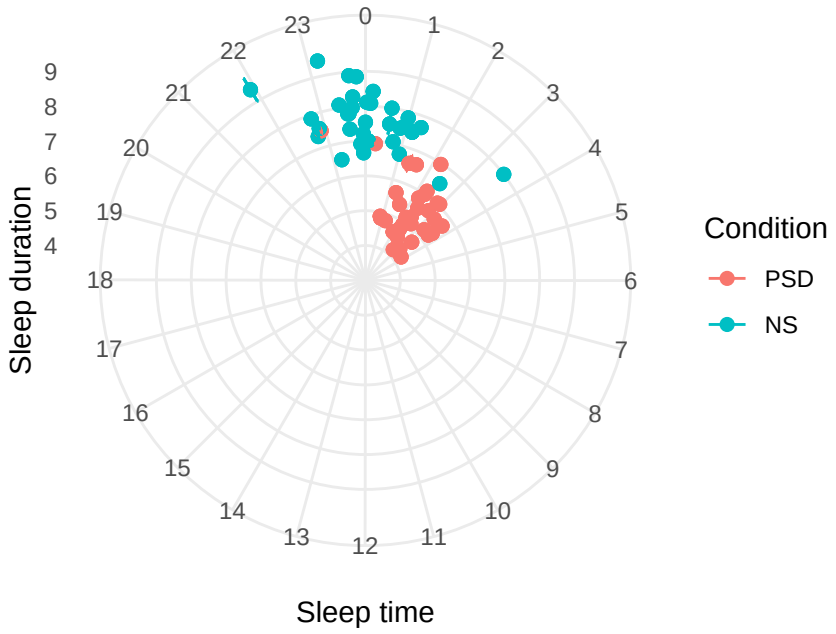
summarizations than actual plots. In particular, when the data points cluster on or near each other, as the circle plot transform the layout of the plot in such a way as to make it difficulty to read or interpret. With this, we can proceed to make it circular.

```
sleep |>
  group_by(sleepdep) |>
  mutate(sd_length = length(sleepdep)) |> ungroup() |>
  group_by(subj, sleepdep) |>
  mutate(
    # Mean sleep duration
    duration = mean(sleep_duration, na.rm=T),
    duration_sd = sd(sleep_duration, na.rm=T),
    duration_se = sd(sleep_duration, na.rm=T) / sqrt(sd_length),
    # mean sleep time over participant by condition
    sleep = mean(sleep_time_cum, na.rm=T),
    sleep = ifelse(sleep>=24, sleep-24, sleep),
        # could add clock time here
    sleep_sd = sd(sleep_time_cum, na.rm=T),
    sleep_se = sd(sleep_time_cum, na.rm=T) / sqrt(sd_length),
    # mean wake time over participant by condition
    waking = mean(last_awaking_fix, na.rm=T),
    waking_sd = sd(last_awaking_fix, na.rm=T),
    waking_se = sd(last_awaking_fix, na.rm=T) / sqrt(sd_length),) |>
  unique() |>
  ggplot(aes(x=sleep, y=duration, col=sleepdep))+
  geom_point(size=2)+
  geom_errorbar(aes(ymin=duration-duration_se, ymax=duration+duration_se))+
  coord_polar() -> sobj1
sobj1
```



Now we can start by making the circle plot more neat. Firstly, the data points all seem to cluster in around 2-3 hours, at least for the PSD condition. We can change this by placing an invisible point closer to the center, which will increase the distance a bit. Moreover, we can change the sleep to a 24 hour clock, indicating when participants went to sleep.

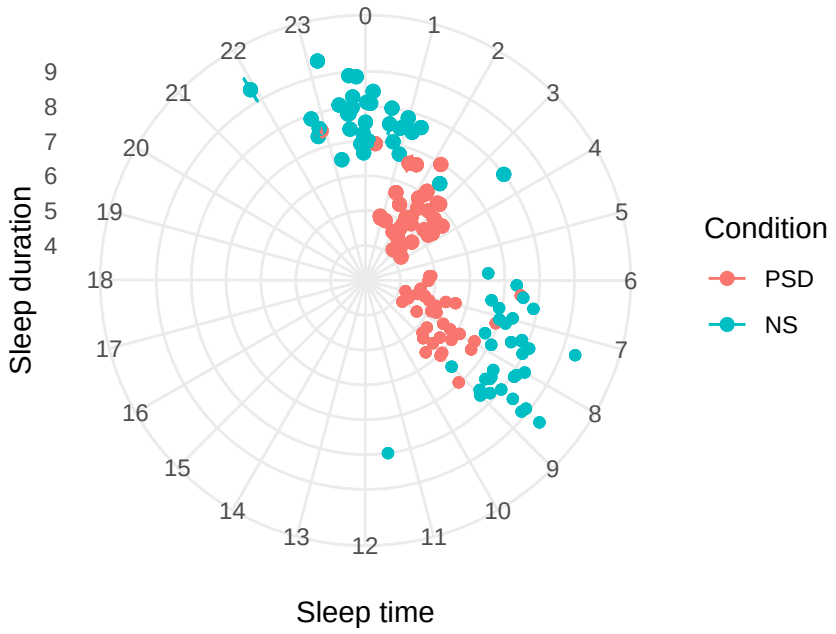
```
sobj1 +
  geom_point(aes(x=1,y=3), alpha =0)+ # some distances from the center
  scale_x_continuous(breaks = seq(0,24,1), limits = c(0,NA), minor_breaks = F)+ # Limits a
  scale_y_continuous(breaks = seq(4,10,1))+
  theme_minimal()+
  labs(x="Sleep time", y="Sleep duration", fill="Condition", col="Condition") -> sobj2
sobj2
```



In this plot, the x-axis represents the time at which participants fell asleep, while the y-axis represents the amount of sleep participants received. From the plot, we can clearly see that during the NS condition, participants went to sleep earlier and slept for longer as compared to the PSD condition. During the PSD condition, participants went to sleep later, and also slept shorter (indicated with points closer to the center.

We can add the wake time and connect these with lines, (but this is probably messy…)

```
sobj2 +
  geom_point(aes(x = waking, y=duration, col=sleepdep))
```



checking the dimensions of the df (indicate the reduction in data points.)

```
# slp |> #dim()
#   filter(subj, sleepdep) |> dim()
#   group_by(subj, sleepdep) |> dim()
```

```
# # dont make dis hard for me, thx
#
# # Join diary & data
# sleep_diary |>
#   mutate(subj=as.numeric(subj),
#          sleepdep = ifelse(pre_control==1, "control",
#                            ifelse(pre_sleepdep==1, "SD", NA))) |>
#           #' Add common var
#   left_join(data.probe, # behaviour data
#           by=c("subj", "sleepdep")) |>
#   mutate(sleepdep = factor(
#     fct_recode(sleepdep, NS="control", PSD="SD"), levels = c("PSD","NS"))) -> sleep_and_
#
# # data filtering & changes
# sleep_and_data |>
```

```
#   filter(!(is.na(sleepdep))) -> slp
```