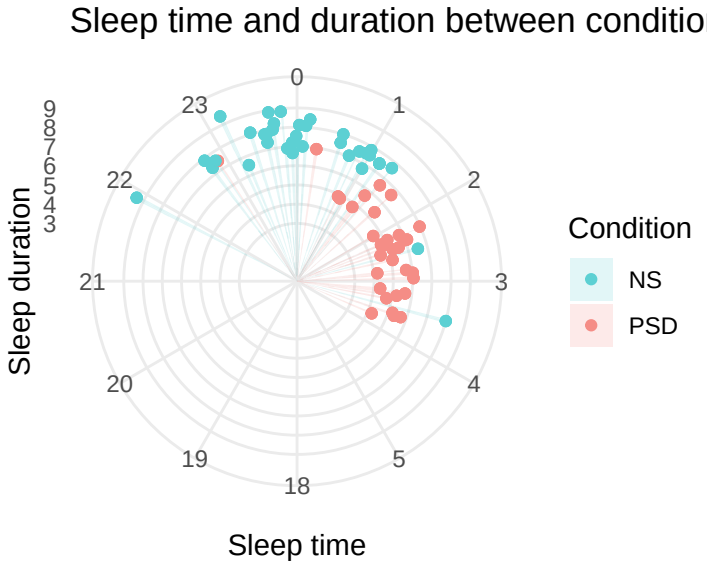
Circle Plot: sleep deprivation



leep time (x). NS = normal sleep, PSD = partial sleep deprivation.
ars represents sleep duration, and x location indicate sleep time.
ccording to clock time displayed from 18 (evening) to 6 (morning).

Here I will further investigate the use of circle plot by focusing on sleep times in a study on sleep deprivation. In a similar way to the seasonal data, sleep can be plotted on a circular plot which can correspond neatly to clock time, as we will also see. Later, I discuss some thoughts about circle plots derived from this investigation and the previous. Essentially, circular plots are good at visualizing summarized data, as well as differences between groups. Moreover, this can be meaningfully displayed according to clock time. Parts of which a normal visualization can miss out on. However, greatly benefiting by being easier to read each data point.

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

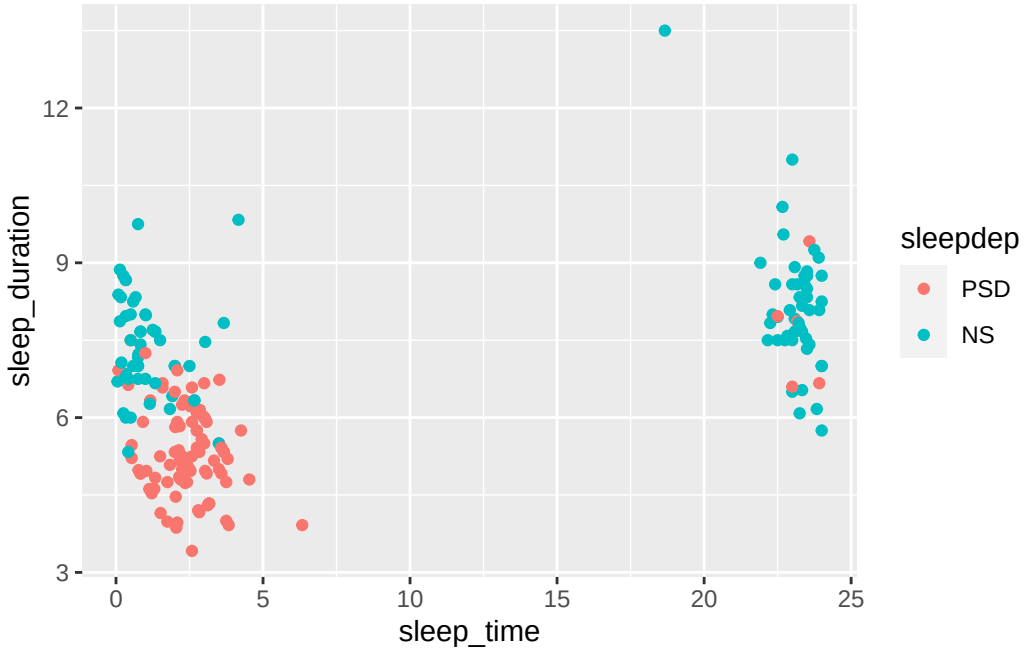
Before we start, some minor changes to the data frame will be done to make the plots look better and make them easier to work with.

```
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.

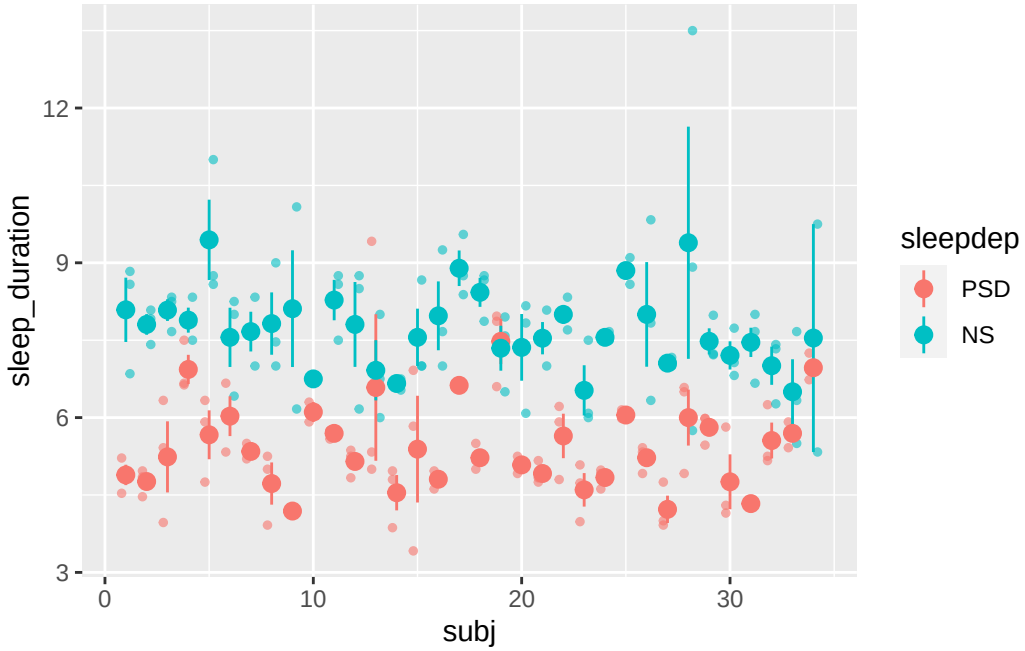
```
sleep |>
  ggplot(aes(x=subj, y=sleep_duration, col=sleepdep))+
  geom_point(position=position_dodge(.8), size=1, alpha=.6)+
  stat_summary()
```

```
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()`).
```

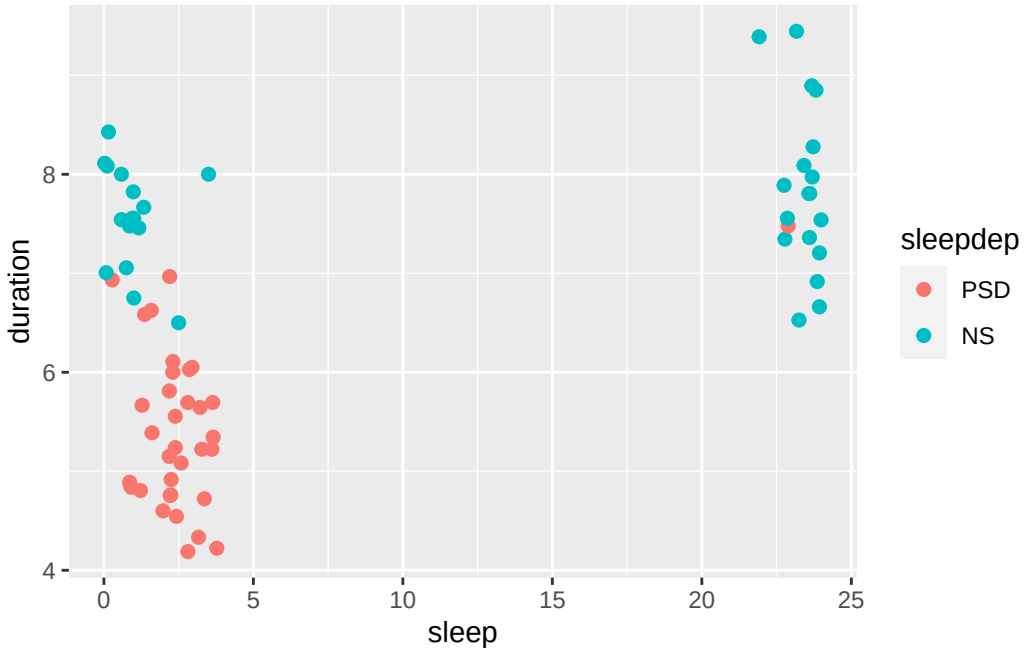
```
Warning: Removed 1 rows containing missing values (`geom_segment()`).
```



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)
```

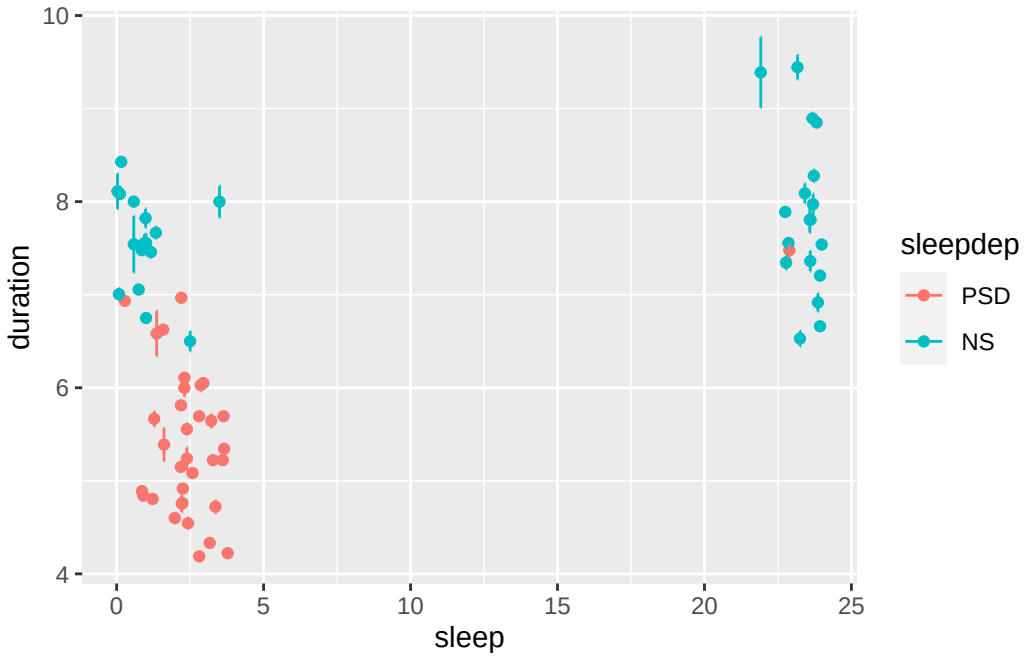


This result in a plot that seem to contain more of what I am looking for. Another thing that could be added to the plot is some uncertainty relating to the data. For instance, by using standard error, standard deviation or confidence intervals. For now I will stick with standard error, but this can easily 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=1.5)+
  geom_errorbar(aes(ymin=duration-duration_se, ymax=duration+duration_se))
```



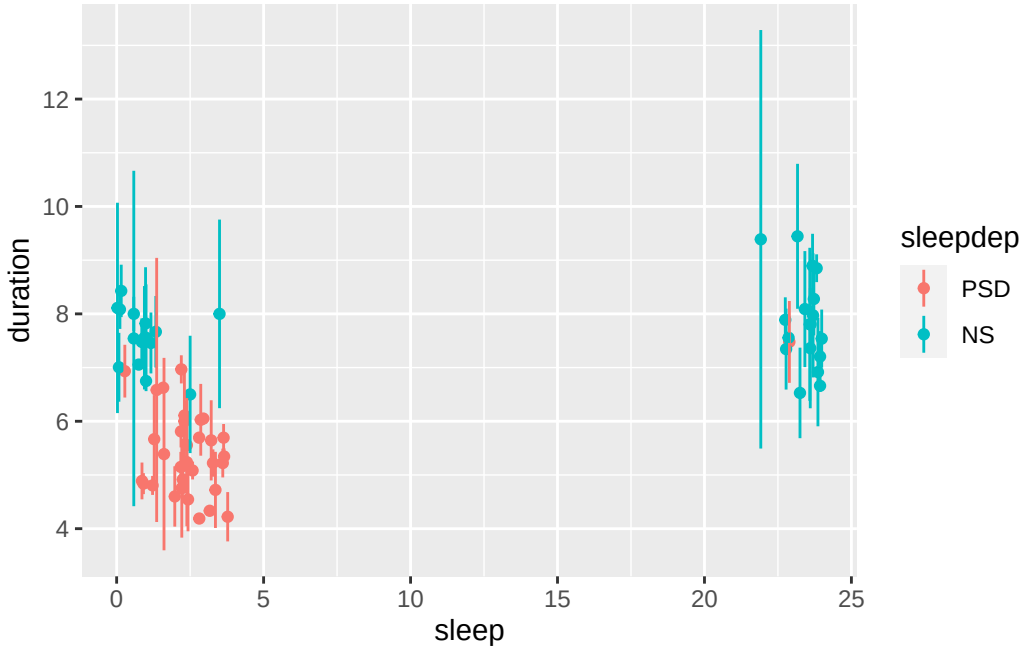
Here I improved upon the data transformation and included a calculation of the standard error (SE) and standard deviation (SD). This will be used to indicate some uncertainty in our estimate or indicate the variation of the data. As noted previously, circle plots are more apt at visualizing summarized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

Before we move on, I would like to quickly investigate whether to stick with SD or SE. To me SD appear to be more informative as it indicate, to some degree, the width of the data. However, exactly why the width of the data is important or interesting may be more difficult to justify. In our case, the SD indicate variation in sleep time. Since we are estimating participants sleep duration, their SD should (hopefully) be rather small, indicating more consistent sleep duration. On the other hand, visualizing SE would indicate the error of the estimate, which also partially indicate that the uncertainty is wider, although it does not really indicate just how wide - in my opinion. To do this, we simply change the calculated "_se" to "_sd" in the "linerange" argument.

```
s_sleep |>
  ggplot(aes(x=sleep, y=duration, col=sleepdep))+
  geom_point()+
  geom_linerange(aes(ymin=duration-duration_sd,ymax=duration+duration_sd))
```

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

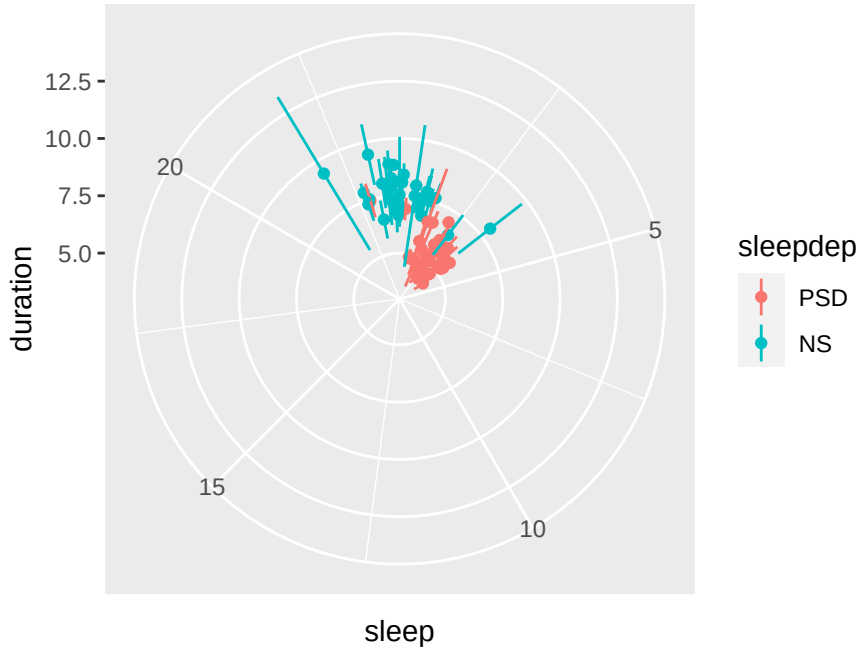


Although I would argue that SD may be more informative, it adds a lot of overlap with our

data estimates. Which does not bode well for visualizing the data circularly. Indeed, it is rather difficult to even see all the estimated points in this plot.

```
s_sleep |>
  ggplot(aes(x=sleep, y=duration, col=sleepdep))+
  geom_point()+
  geom_linerange(aes(ymin=duration-duration_sd,ymax=duration+duration_sd))+
  coord_polar()+
  geom_blank(aes(y=3))
```

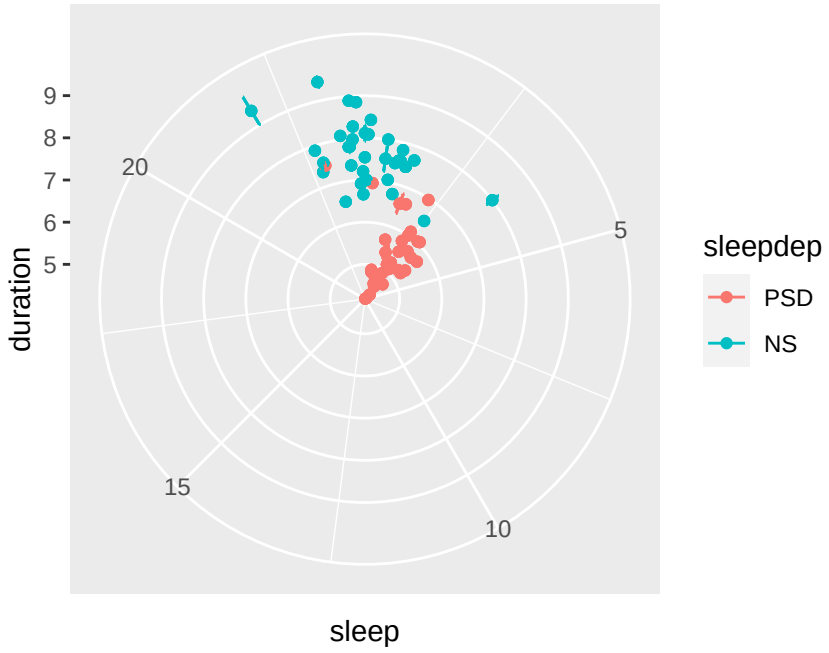
```
Warning: Removed 1 rows containing missing values (`geom_segment()`).
```



Indeed, with the SD in the circular plot, it turns it to a mess. It even seems like some points are lost in the soup of the PSD sleep times. Thus, for further investigation I will stick with SE, which makes the data points more readable, unless otherwise noted.

```
sleep |>
  group_by(sleepdep) |>
  mutate(sd_length = length(sleepdep)) |> ungroup() |>
  group_by(subj, sleepdep) |>
  mutate(
```

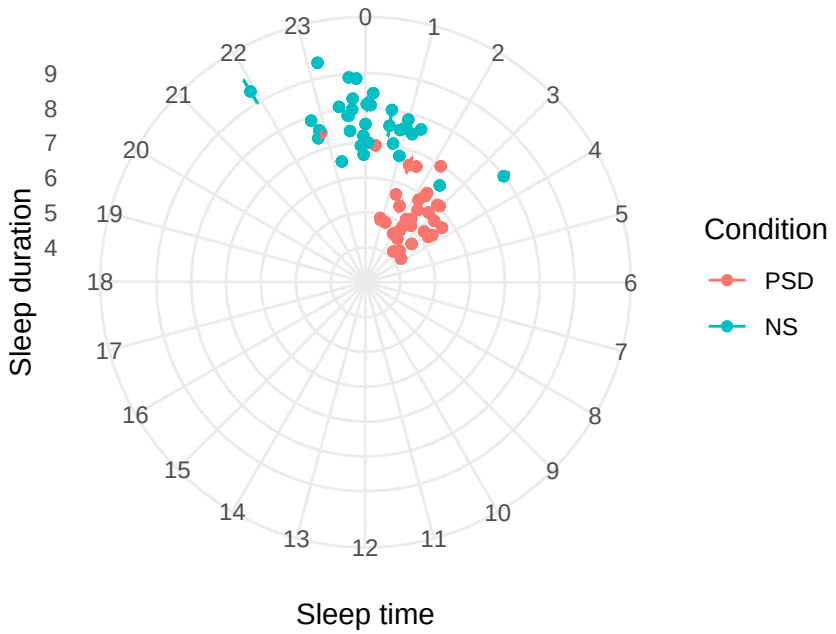
```
    # Mean sleep duration
    duration = mean(sleep_duration, na.rm=T),
    duration_sd = sd(sleep_duration, na.rm=T),
    duration_se = sd(sleep_duration, na.rm=T) / sqrt(sd_length),
    # mean sleep time over participant by condition
    sleep = mean(sleep_time_cum, na.rm=T),
    sleep = ifelse(sleep>=24, 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=1.5)+
  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_blank(aes(x=1,y=3))+ # Add an empty element too add 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
```

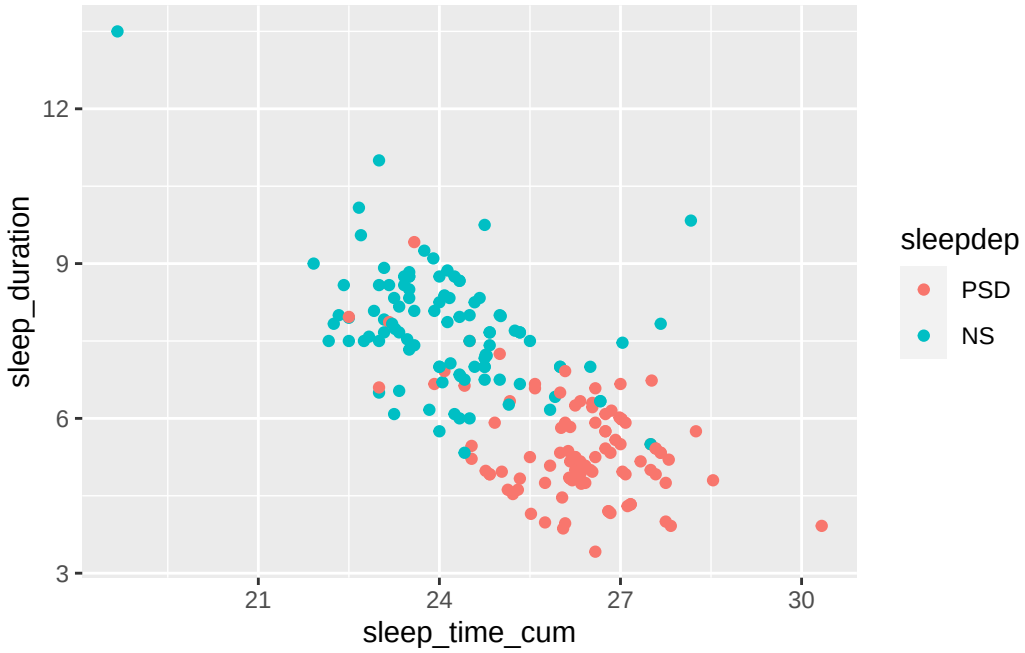


In this plot, the x-axis represents the time at which participants fell asleep, while the y-axis represents the amount of sleep participants received. From the plot, we can clearly see that during the NS condition, participants went to sleep earlier and slept for longer as compared to the PSD condition.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

With the previous plot we are able to approximate when participants went to sleep, and approximately how long they slept. We could possibly make this more clear by using the cumulative sum of the clock instead of a 24 hour clock.

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

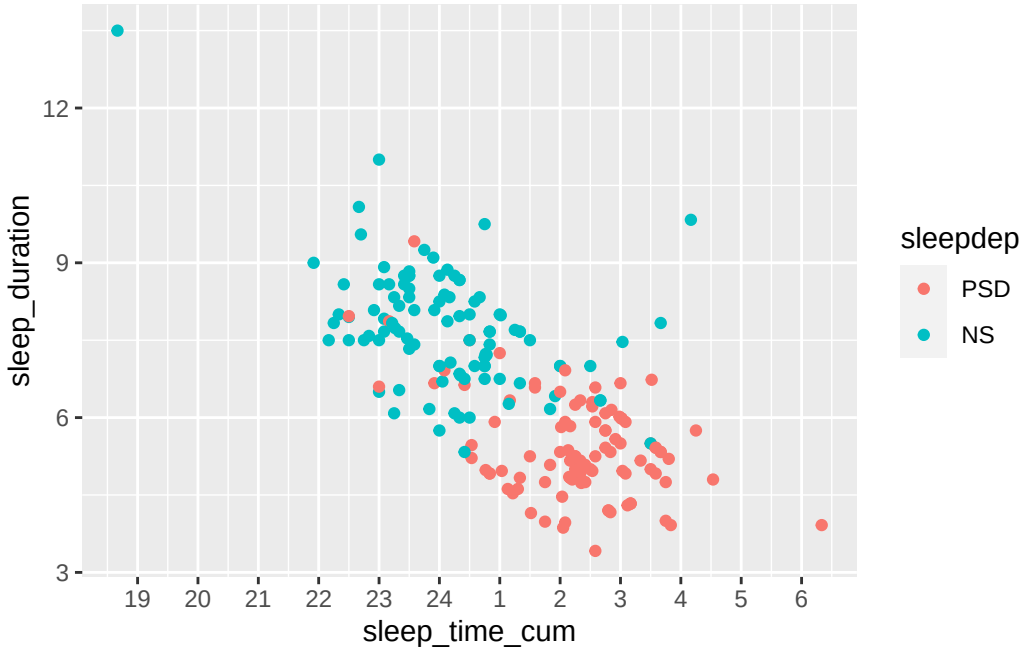
```
Warning: Removed 8 rows containing missing values (`geom_point()`).
```



Firstly, with this plot we can clearly see a downward trend for the sleep duration plotted against the sleep time. Keeping the data in this way may be easier to work with in the future, and we can easily change the appears of the x-axis to the appropriate time (i.e., 0, 1, 2, etc.).

```
sleep |>
  ggplot(aes(x=sleep_time_cum, y=sleep_duration, col=sleepdep))+
  geom_point()+
  scale_x_continuous(breaks=seq(0,31,1), labels=c(seq(0,24,1), seq(1,7,1)))
```

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

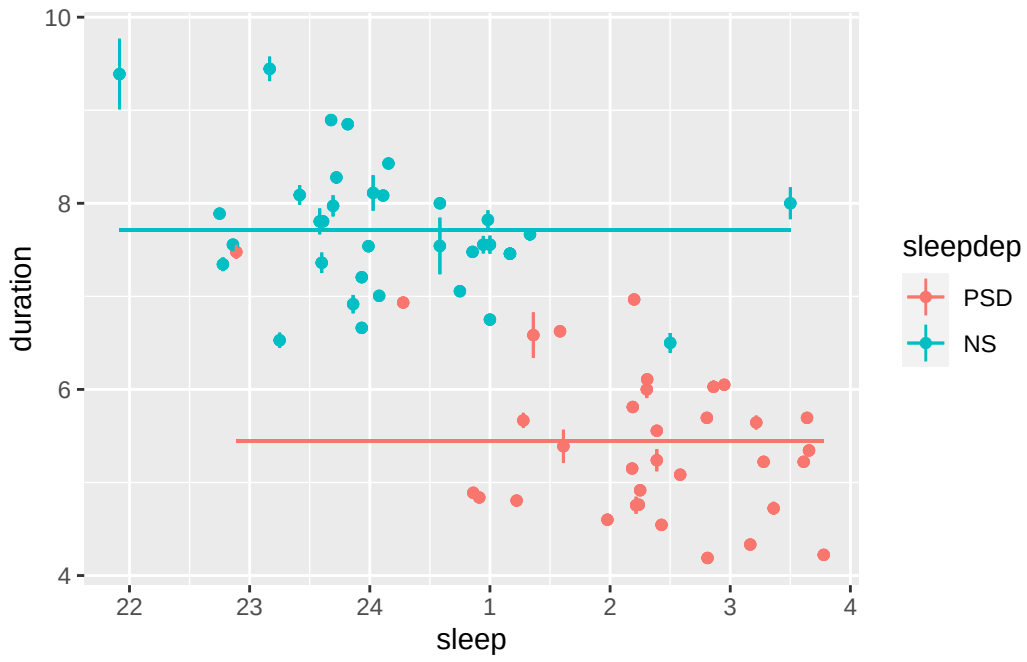


This looks similar to previously (which is good), but it should now be easier to work with.

```
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),
        # 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),
    ) -> s_sleep2
```

```
s_sleep2 |>
  ggplot(aes(x=sleep, y=duration, col=sleepdep))+
  geom_point()+
  geom_linerange(aes(ymin=duration-duration_se, ymax=duration+duration_se, x=sleep))+
  # add NS mean line
  geom_linerange(aes(y=mean(duration), xmin=min(sleep), xmax=max(sleep)),
                 s_sleep2 |> filter(sleepdep=="NS"))+
  # Add PSD mean line
  geom_linerange(aes(y=mean(duration), xmin=min(sleep), xmax=max(sleep)),
                 s_sleep2 |> filter(sleepdep=="PSD"))+
  # fix the x-axis
  scale_x_continuous(breaks=seq(0,31,1), labels=c(seq(0,24,1), seq(1,7,1))
                     ) -> sobj10
sobj10
```

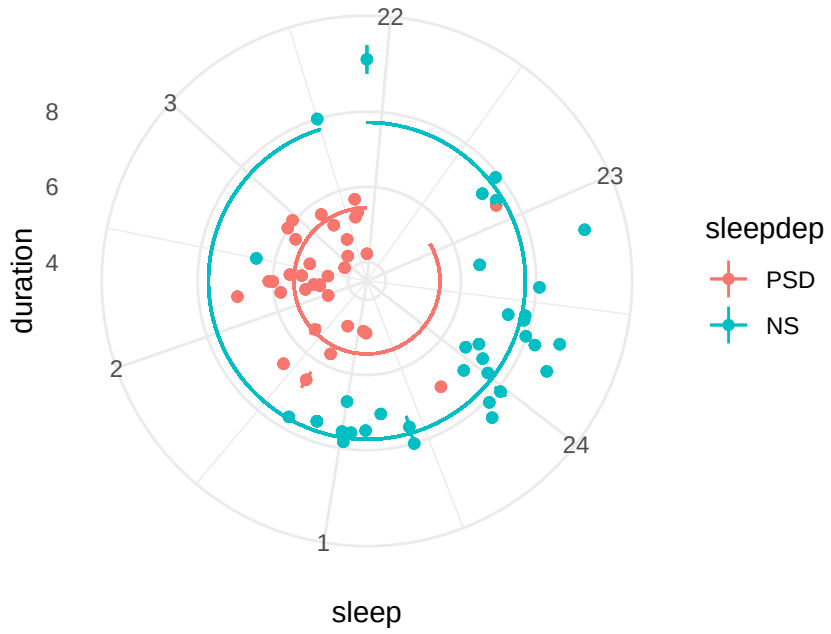
```
Warning: Removed 3 rows containing missing values (`geom_segment()`).
```



To this end, it is now a rather easy to add a mean line of sleep duration for each conditions. With this, we can clearly see that participants, in general, slept less during the PSD condition than the NS condition.

```
sobj10 +
  coord_polar() +
  theme_minimal()+
  geom_blank(aes(y=3.5))
```

```
Warning: Removed 3 rows containing missing values (`geom_segment()`).
```



A benefit with this plot is that all the data points are easier to locate. However, it is now less obvious what the time is, as compared to the previous plots, in which the x-axis corresponded to clock time. One of the benefits of using a circular plot, is to visualize data in an informative or intuitive way. In the above plot, we need to constantly check up against the x-axis to understand which time we are at. This is couple of times easier to do with the normal plot. However, if we turn the location of the circle plot to correspond to the normal clock time, maybe things will turn more neatly.

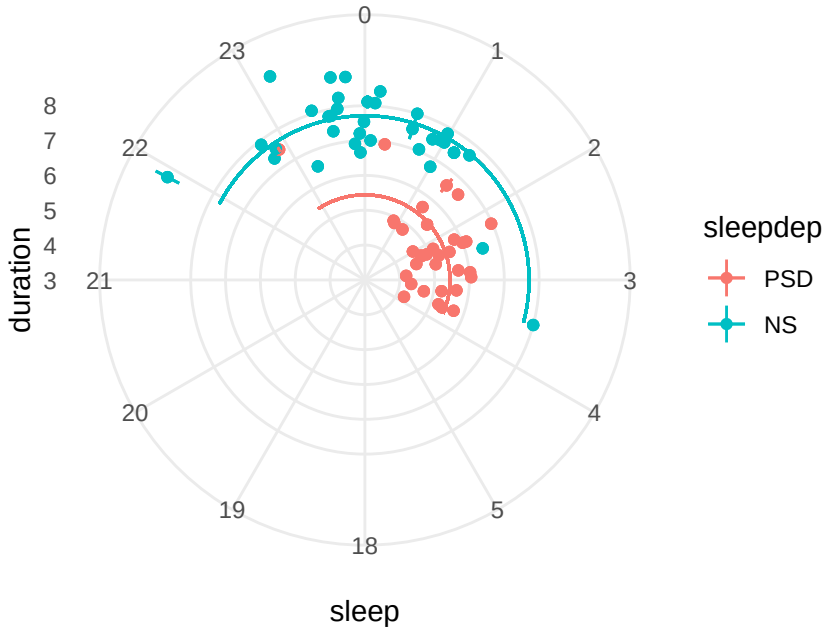
```
sobj10 +
  coord_polar(start = -pi)+ # Equal distance between each side (center @ 0)
  geom_blank(aes(x=30,y=3))+ # Latest (empty) point
  geom_blank(aes(x=18,y=3))+ # Earliest (empty) point
  scale_x_continuous(breaks=seq(0,29,1), labels=c(seq(0,23,1), seq(0,5,1)), minor_breaks =
  scale_y_continuous(breaks=seq(3,8,1))+
```

```
  theme_minimal() -> sobj11
```

```
Scale for x is already present.
Adding another scale for x, which will replace the existing scale.
```

```
sobj11
```

```
Warning: Removed 3 rows containing missing values (`geom_segment()`).
```

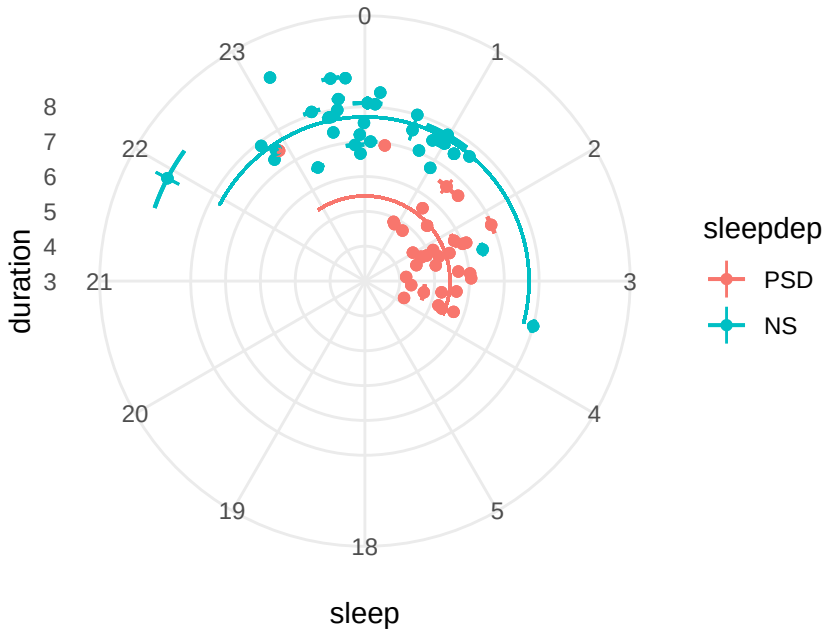


This plots does indeed look much better. Moreover, we are meaningfully visualizing clock time, at least to some degree, within a 12-hour span. From 18 to 6, which should (and in our case do) capture all sleep estimates in a visually pleasing way. Moreover, the layout of the circle correspond to a large extent with the plot containing all hours of the day

One missing feature is that we do not see the uncertainty in each participants sleep time. This can be done by adding a width element to each estimate, which correspond to the uncertainty of the sleep time estimate. However, this may increase clutter of the plot, especially since the data points clutter together more in a circle plot.

```
sobj11 +
  geom_linerange(aes(y=duration, xmin=sleep-sleep_se, xmax=sleep+sleep_se), linewidth=.7)
```

```
Warning: Removed 3 rows containing missing values (`geom_segment()`).
Removed 3 rows containing missing values (`geom_segment()`).
```

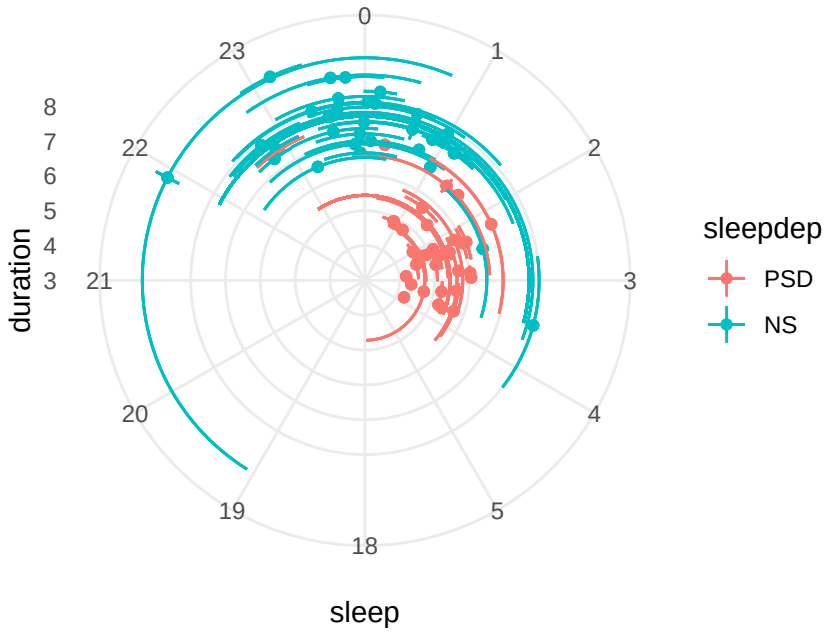


In this plot, we also get some uncertainty on the x-axis, corresponding to the sleep time. However, it does add a bit more clutter, making it hard to see some of the data points.

I do not believe visualizing with SD will be more pleasing, but it is nonetheless something worth checking out.

```
sobj11 +
  geom_linerange(aes(y=duration, xmin=sleep-sleep_sd, xmax=sleep+sleep_sd))
```

```
Warning: Removed 3 rows containing missing values (`geom_segment()`).
Removed 3 rows containing missing values (`geom_segment()`).
```



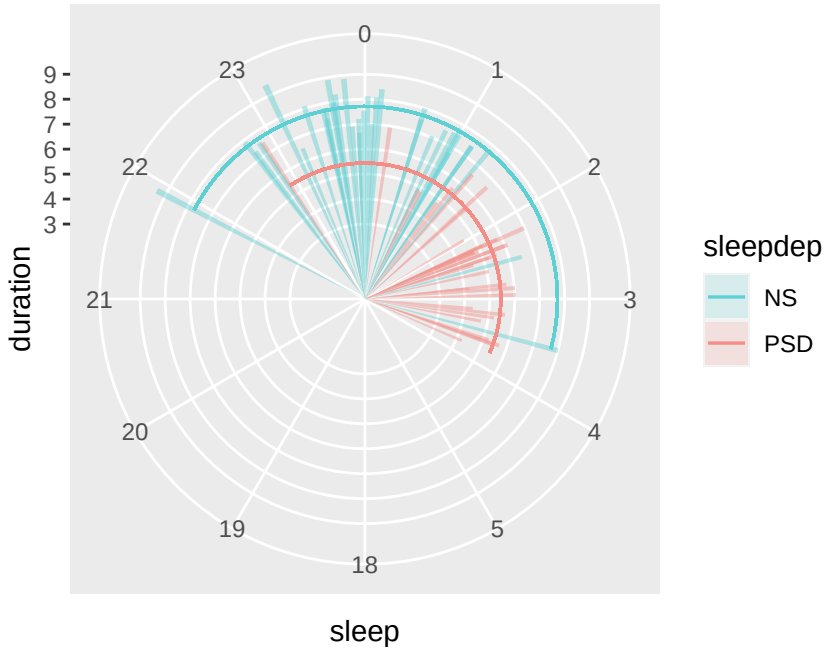
Arguably, this plot is more difficult to read, and, moreover, visualizing SE and SD is misleading. Nevertheless, I find it rather informative in that we can see the variation of the sleep times. Which is quite large for some individual, while being more consistent. Although SE also indicate some uncertainty, it does not indicate the width of the data. For the future, I will stick with SE on the x-axis to be consistent and not generate too much clutter.

One alternative could be to visualize with a bars, such that each point receives its own bar length corresponding to the average sleep duration.

```
s_sleep2 |>
  mutate(sleepdep=as.character(sleepdep)) |> #unfactor
  ggplot(aes(x=sleep, y=duration, fill=sleepdep, group=sleepdep))+
  geom_col(position=position_dodge(), width = 0.05, alpha=.18)+
  scale_fill_manual(values = c("#5cd0d3","#f58d86"))+ # change the colour
  # Add PSD mean line
  geom_linerange(aes(y=mean(duration), xmin=min(sleep), xmax=max(sleep), col=sleepdep),
                 s_sleep2 |> filter(sleepdep=="PSD"))+
  # add NS mean line
  geom_linerange(aes(y=mean(duration), xmin=min(sleep), xmax=max(sleep), col=sleepdep),
                 s_sleep2 |> filter(sleepdep=="NS"))+
  scale_colour_manual(values = c("#5cd0d3","#f58d86"))+ # change the colour
  coord_polar(start = -pi)+ # Equal distance between each side (center @ 0)
```

```
  geom_blank(aes(x=30,y=3))+ # Latest (empty) point
  geom_blank(aes(x=18,y=3))+ # Earliest (empty) point
    #These change the display of the clock time.
  # fix the x-axis (or clock time)
  scale_x_continuous(breaks=seq(0,29,1), labels=c(seq(0,23,1), seq(0,5,1)), minor_breaks =
  scale_y_continuous(breaks=seq(3,10,1)) -> sobj20
sobj20
```

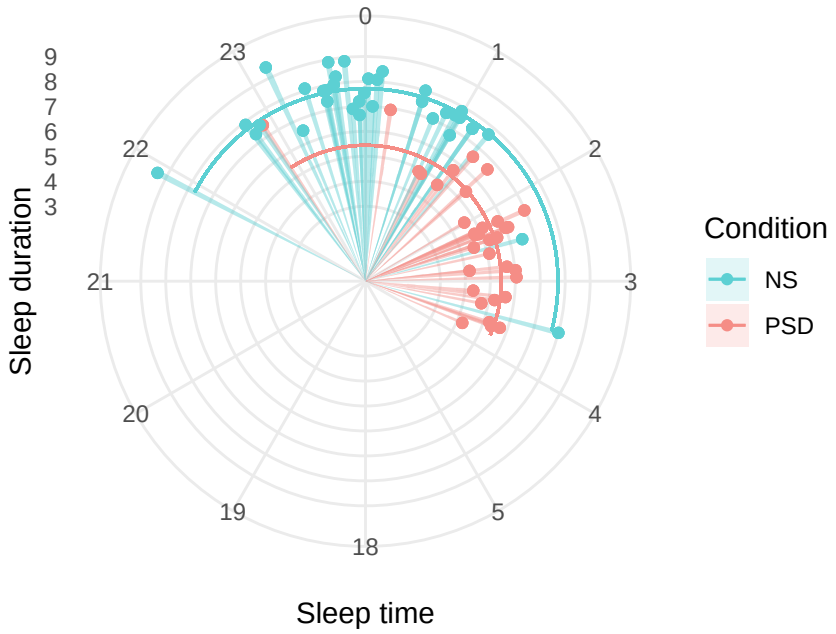
```
Warning: `position_dodge()` requires non-overlapping x intervals
```



In this plot, it is perhaps more obvious that the sleep duration in the PSD condition is reduced as compared to the NS condition. Moreover, the circle plot does fit this type of visualization more than plotting dots, as the bars themselves convey information about length. However, it is perhaps most clearly illustrated by a contrast. That is to say, it is still difficult to estimate the exact value of any data point, but it does put the conditions in contrast. Suggesting, perhaps, that circle plots are better at visualizing contrasts than at presenting data. It should be noted that it can appear that the difference between the NS and the PSD condition is exaggerated since the appears of the bars reduced closer to the circle, as compared to the NS bars. This may, however, serve a purpose as well. Our research, and others, indicate that small deviation (although ours may be considered medium) of sleep amount can have severe impact on performance.

```
sobj20 +
  geom_point(aes(col=sleepdep))+ # these clarify exact data point (& put to contrast)
  theme_minimal()+
  labs(x="Sleep time", y="Sleep duration", fill="Condition", col="Condition")
```

```
Warning: `position_dodge()` requires non-overlapping x intervals
```

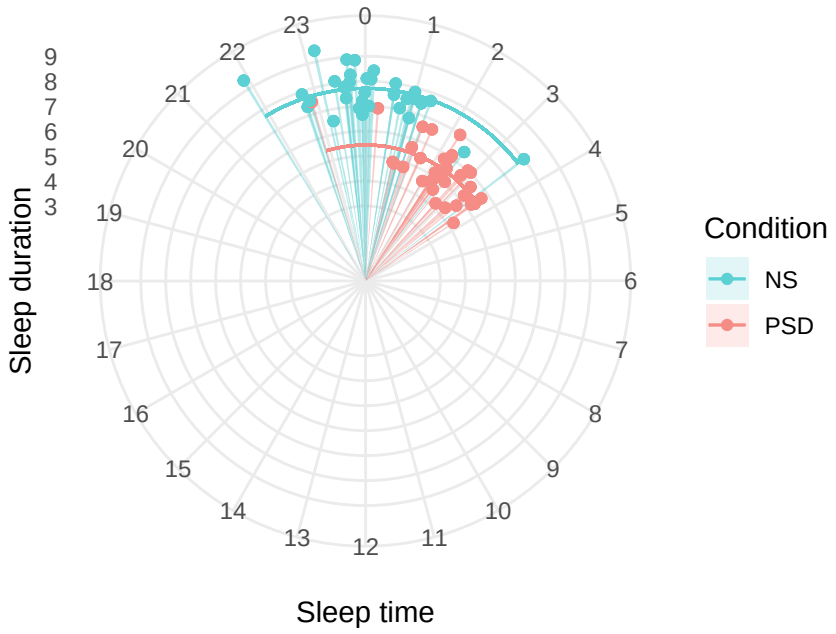


Despite the fact that we leave a fair amount of the plot to nothing (i.e., 4-21), displaying the data in this way is at least meaningful in relation to a clock (at least to some degree). However, this way is at least better than visualizing all 24 hours of the day (see below), since that would create even more non-data ink. Although a plot visualizing all the hours of the day, could benefit by also indicating the wake time of the related sleep sleep times.

```
sobj20 +
  geom_point(aes(col=sleepdep))+ # these clarify exact data point (& put to contrast)
  theme_minimal()+
  labs(x="Sleep time", y="Sleep duration", fill="Condition", col="Condition")+
  geom_blank(aes(x=6))+
  coord_polar(start = 1.57)
```

```
Coordinate system already present. Adding new coordinate system, which will
replace the existing one.
```

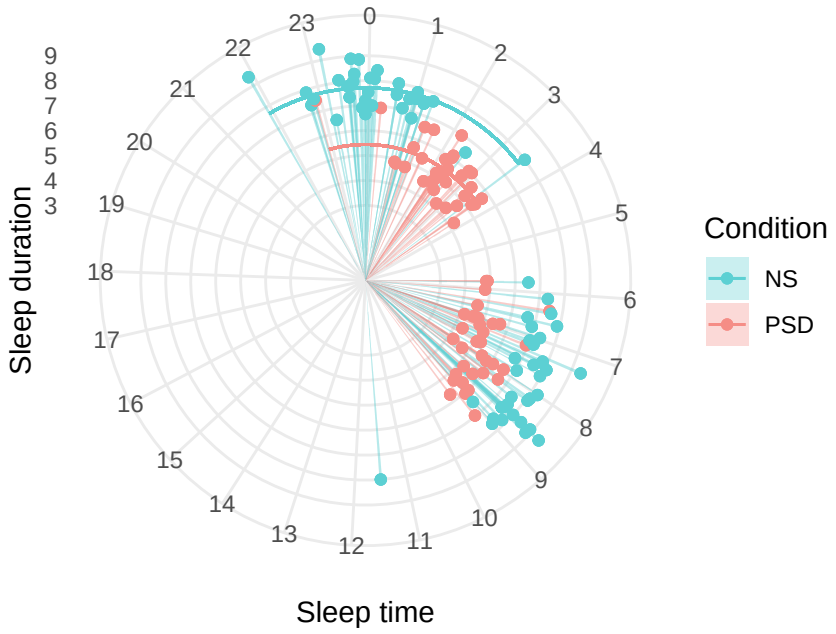
```
Warning: `position_dodge()` requires non-overlapping x intervals
```




```
sobj20 +
  geom_point(aes(col=sleepdep))+ # these clarify exact data point (& put to contrast)
  theme_minimal()+
  labs(x="Sleep time", y="Sleep duration", fill="Condition", col="Condition")+
  geom_blank(aes(x=6))+
  coord_polar(start = 1.57)+
  geom_col(aes(x=waking), position = position_dodge(), width=.05, alpha=.18)+
  geom_point(aes(x=waking, col=sleepdep))
```

```
Coordinate system already present. Adding new coordinate system, which will
replace the existing one.
```

```
Warning: `position_dodge()` requires non-overlapping x intervals
`position_dodge()` requires non-overlapping x intervals
```



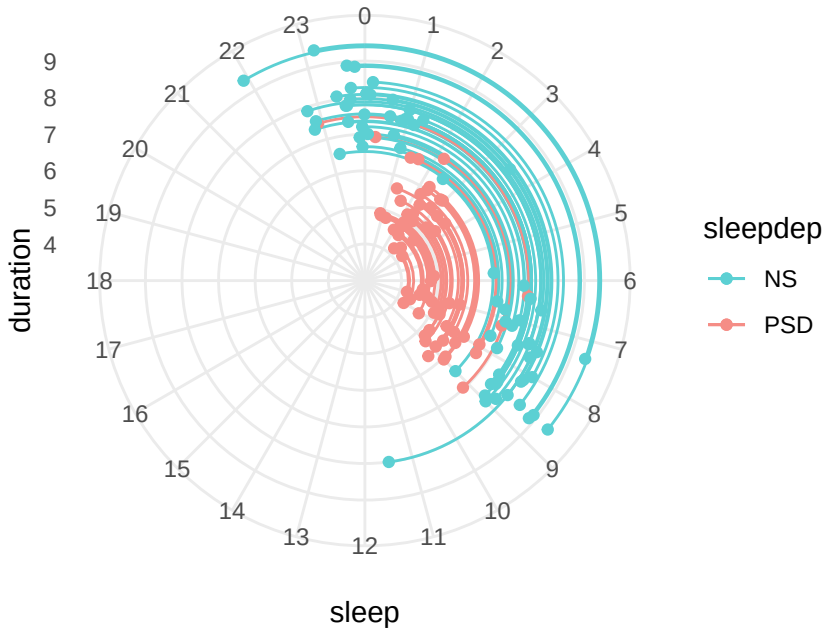
In this plot we can also see the wake time at around 8 o'clock. Although this should be visualized with a different colour to not mix the sleep time and wake time. For now I will not distinguish between the two as I am mostly interest in visualizing sleep times. Although some investigation into the relationship between sleep time and wake time is warranted.

```
s_sleep |>
  mutate(sleepdep=as.character(sleepdep)) |> #unfactor
  ggplot(aes(x=sleep, y=duration, fill=sleepdep, col=sleepdep, group=sleepdep))+
  geom_linerange(aes(xmin=ifelse(sleep<5,sleep,NA), xmax=waking, y=duration))+
    # if after 24 then draw to waking time
  geom_linerange(aes(xmin=ifelse(sleep>19,sleep,NA), xmax=24, y=duration))+
    # if later than 19, draw to 24
  geom_linerange(aes(xmin=ifelse(sleep>19,0,NA), xmax=waking, y=duration))+
    # if later than 19 start from 0 and draw to waking
  geom_point()+ # sleep times
  geom_point(aes(x=waking, col=sleepdep))+ # Waking times
  scale_colour_manual(values = c("#5cd0d3","#f58d86"))+ # change the colour
  geom_blank(aes(x=1,y=3))+ # Add an empty element too add some distances from the center
  coord_polar()+
  scale_x_continuous(breaks = seq(0,23,1), limits = c(0,NA), minor_breaks = F)+ # Limits a
  scale_y_continuous(breaks = seq(4,10,1))+
  theme_minimal() -> sobj30
```

```
sobj30
```

```
Warning: Removed 19 rows containing missing values (`geom_segment()`).
```

```
Warning: Removed 49 rows containing missing values (`geom_segment()`).
Removed 49 rows containing missing values (`geom_segment()`).
```



In this plot, we can see both the sleep times and wake times connected with a x-axis line at the height of the sleep duration. This line connects the sleep time and wake time point which illustrates that during the NS condition participants tended to sleep longer, by going to sleep earlier but nonetheless waking up at the same time. To further add to this point, we can add to this mean lines indicating the mean sleep time and wake time for the different conditions.

```
s_sleep2 |>
  group_by(sleepdep) |>
  summarize(m_sleep = mean(sleep),
            m_wake = mean(waking))
```

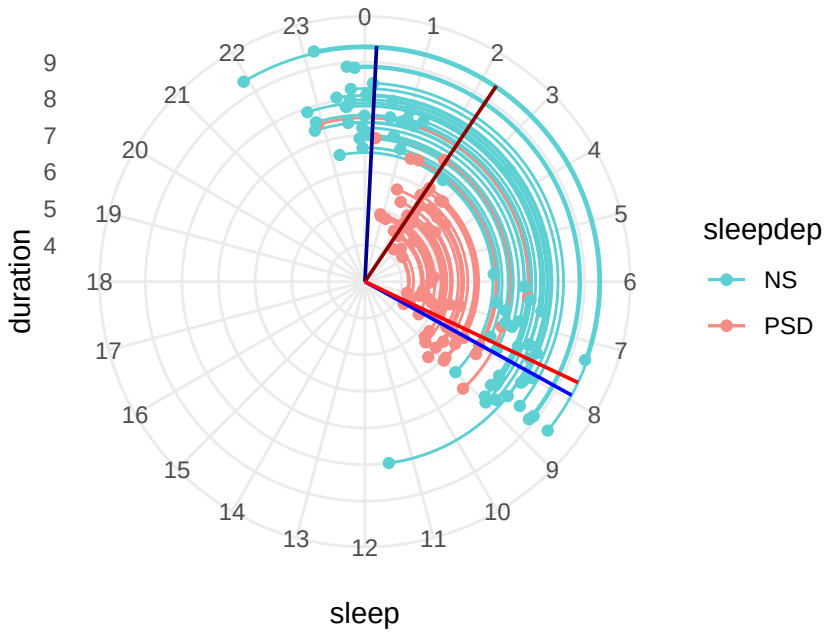
```
# A tibble: 2 x 3
  sleepdep m_sleep m_wake
```

```
  <fct>      <dbl>  <dbl>
1 PSD         26.3   7.69
2 NS          24.2   7.92
```

```
sobj30 +
  # NS
  geom_vline(xintercept=24.18931-24, col="darkblue", linewidth=.65)+ #sleep
  geom_vline(xintercept=7.918789, col="blue", linewidth=.65)+ #wake
  #PSD
  geom_vline(xintercept=26.25842-24, col="darkred", linewidth=.65)+ # sleep
  geom_vline(xintercept=7.688480, col="red", linewidth=.65) -> sobj31 # wake
sobj31
```

```
Warning: Removed 19 rows containing missing values (`geom_segment()`).
```

```
Warning: Removed 49 rows containing missing values (`geom_segment()`).
Removed 49 rows containing missing values (`geom_segment()`).
```



In this plot we can see the average sleep times (dark blue, darkred) and wake time (blue, red) for the NS and PSD respectively. As we can clearly see, participants did go to sleep later

during the PSD condition, but still woke up around the same time. It is perhaps a little less obvious with the lines connecting each data point (see below). Although the lines makes it more difficult to see each data point, it does tell the story of the difference between the conditions. We can see that virtually all the participants in the NS condition is above the lines of the PSD conditions. This is a good sign, as it indicates that participants slept more in the NS condition as compared to the PSD condition. However, it is again important to point out that it is rather difficult to read the exact value of each data point. Indeed, as can be seen (two) below, plotting data in a normal way makes it way more easier to read the sleep duration. It is way more intuitive to read the values from a normal plot than a circle plot.

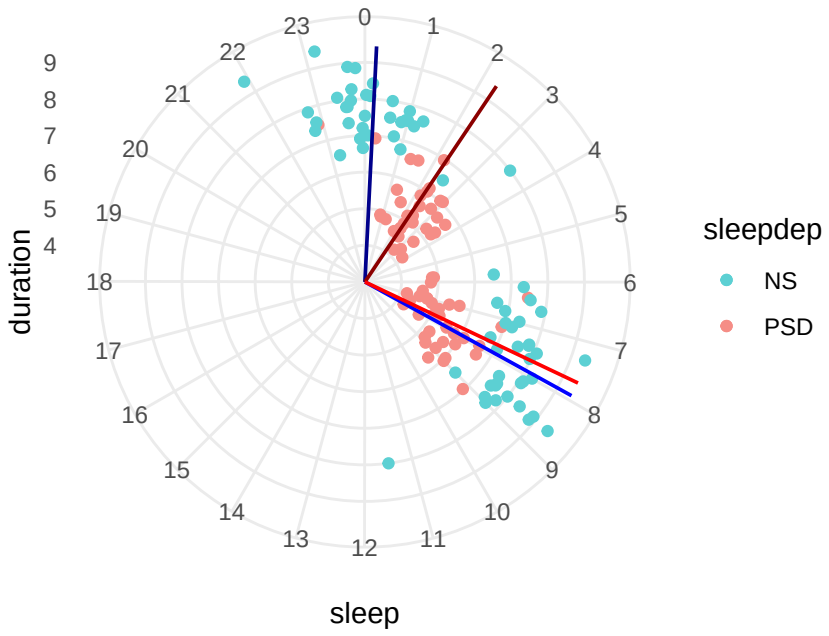
It stands to reason, then, that circle plot may convey an essence but missing details, while a normal plot can convey the exact information of each data point, but perhaps not as neatly. Although the last plot is not as neat as the circle plot, I have not put effort into making it more neat (and will not currently). However, visualizing in this way, seems to be a waste of ink, as the same information is conveyed along the lines from sleep time to the wake time. Indeed, the second plot I created (at the very start) summarize the sleep amount over each participant and displays all the information. However, this plot does not incorporate clock time in any meaningful way.

Depending on what we aim, we can accomplish different things by visualizing with circular plots and normal plots. Indeed, the second plot clearly visualize the sleep amount for each participant across condition, and display each of the data points. In essence, this plot illustrate the sleep duration of each participant across conditions clearly. However, it does not tell us anything about when participants went to sleep and woke up. The latest circle plots, clearly approximates the sleeping times (and waking times) and puts the *conditions* at contrast. In may ways, this is what we want. We want to investigate whether participants slept less during the PSD condition AND whether they went to sleep later as compared to the NS condition.

Thus, circle plots may lose exact information - or at least makes it more difficult to interpret - they can nicely illustrate difference between condition, and they work nicely with clock time. Moreover, circle plots could possible work nicely together with things that tend in circular manners - like the clock and, as previously done, as the seasons.

```
s_sleep |>
  mutate(sleepdep=as.character(sleepdep)) |> #unfactor
  ggplot(aes(x=sleep, y=duration, fill=sleepdep, col=sleepdep, group=sleepdep))+
  geom_point()+ # sleep times
  geom_point(aes(x=waking, col=sleepdep))+ # Waking times
  scale_colour_manual(values = c("#5cd0d3","#f58d86"))+ # change the colour
  geom_blank(aes(x=1,y=3))+ # Add an empty element too add some distances from the center
  coord_polar()+
  scale_x_continuous(breaks = seq(0,23,1), limits = c(0,NA), minor_breaks = F)+ # Limits a
  scale_y_continuous(breaks = seq(4,10,1))+
```

```
  theme_minimal()+
   # NS
  geom_vline(xintercept=24.18931-24, col="darkblue", linewidth=.65)+ #sleep
  geom_vline(xintercept=7.918789, col="blue", linewidth=.65)+ #wake
  #PSD
  geom_vline(xintercept=26.25842-24, col="darkred", linewidth=.65)+ # sleep
  geom_vline(xintercept=7.688480, col="red", linewidth=.65) # wake
```

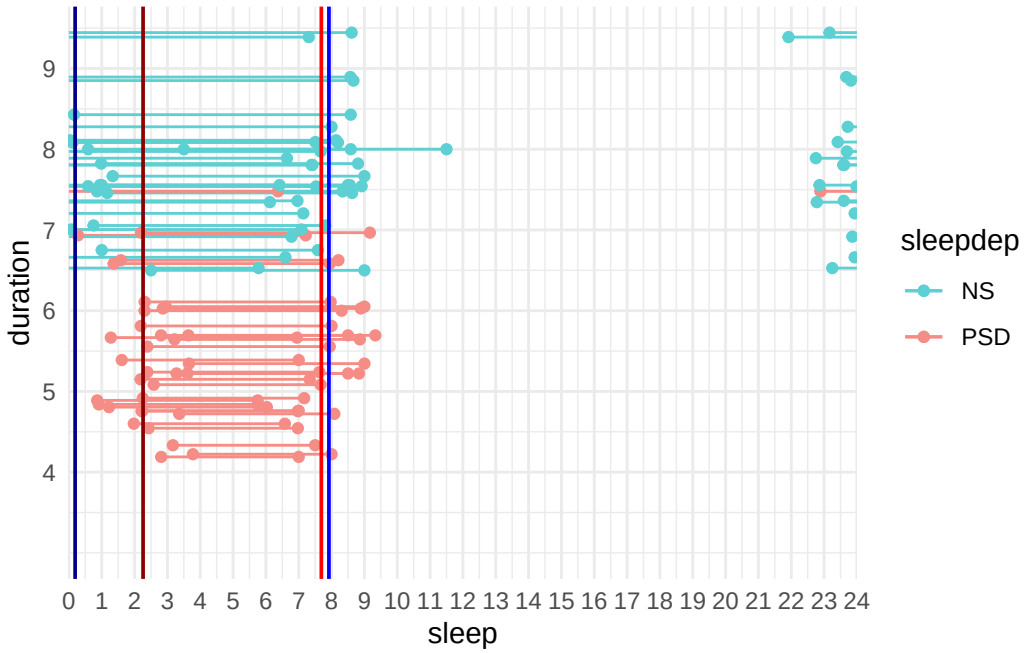



```
sobj31 +
  coord_cartesian()+
  scale_x_continuous(breaks=seq(0,24,1), expand = expansion(0))
```

```
Coordinate system already present. Adding new coordinate system, which will
replace the existing one.
Scale for x is already present.
Adding another scale for x, which will replace the existing scale.

Warning: Removed 19 rows containing missing values (`geom_segment()`).

Warning: Removed 49 rows containing missing values (`geom_segment()`).
Removed 49 rows containing missing values (`geom_segment()`).
```


duration
9
8
7
6
5
4
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
sleep
sleepdep
NS
PSD