Class_Timelog

```
library(ggplot2)
library(dplyr)
```

```
##
## 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(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(ical)
library(wesanderson)
library(scales)
library(lubridate)
```

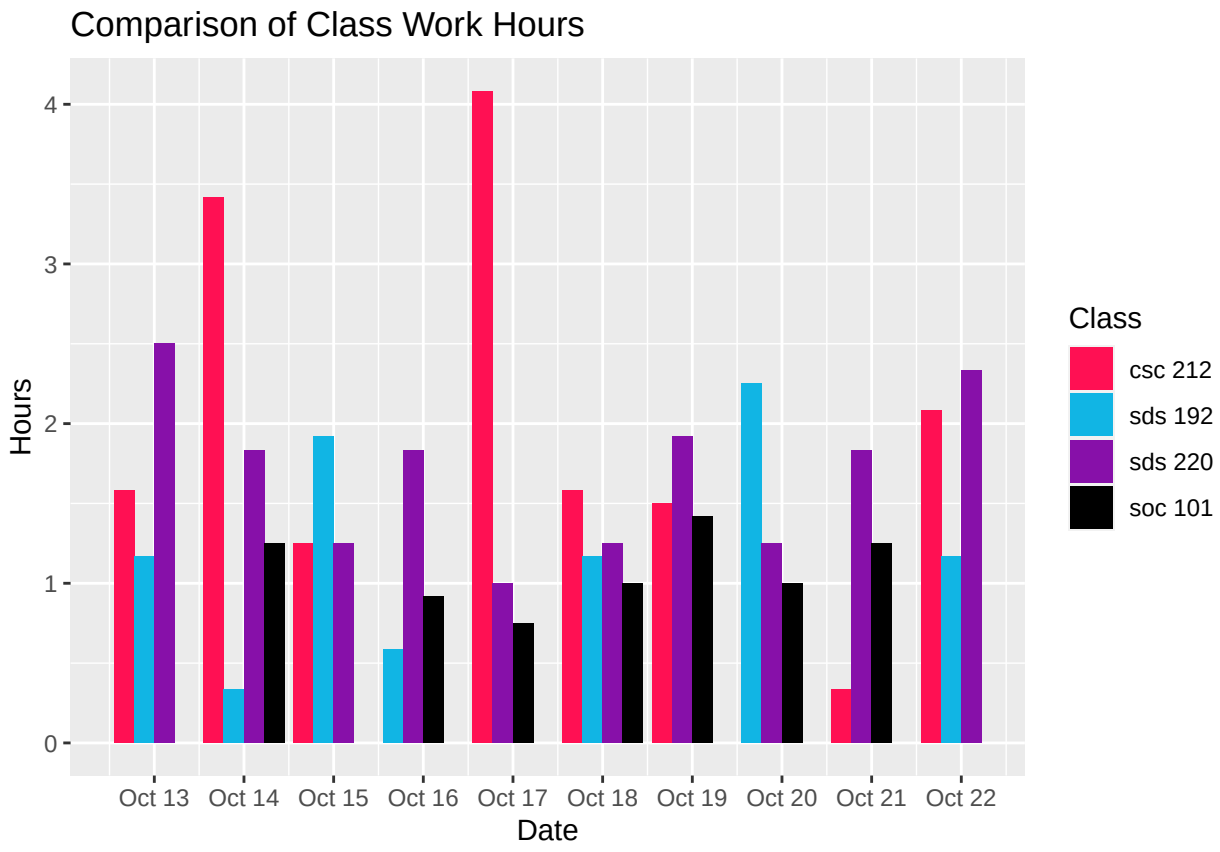
```
calendar_data <- "eek.ics" %>%
  # Use ical package to import into R and then convert to "tibble" data frame format:
  ical_parse_df() %>%
  as_tibble() %>%
  # Use lubridate packge to wrangle dates and times. We'll do this later this semester:
  mutate(
    start_datetime = with_tz(start, tzone = "America/New_York"),
    end_datetime = with_tz(end, tzone = "America/New_York"),
    minutes = end_datetime - start_datetime,
    date = floor_date(start_datetime, unit = "day")
  ) %>%
  # Make calendar entry summary all lowercase:
  mutate(summary = tolower(summary)) %>%
  # Do data wrangling to compute number of minutes and hours:
  group_by(date, summary) %>%
  summarize(minutes = sum(minutes) %>% as.numeric()) %>%
  mutate(hours = minutes/60) %>%
  mutate(date = as.Date(date))
```

```
## `summarise()` has grouped output by 'date'. You can override using the `.groups` argument.
```

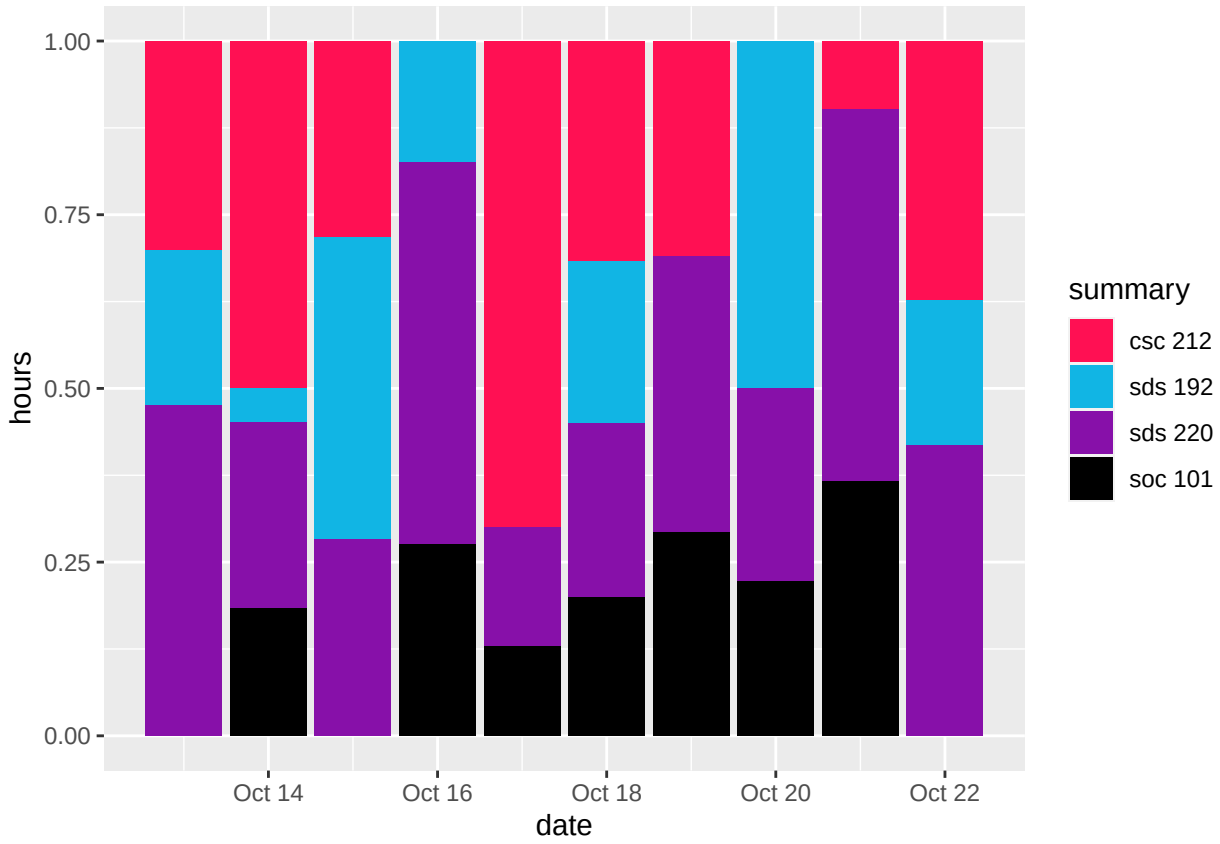
```
  calendar_data <- filter(calendar_data, date > 2000-01-01)
```

```
View(calendar_data)
```

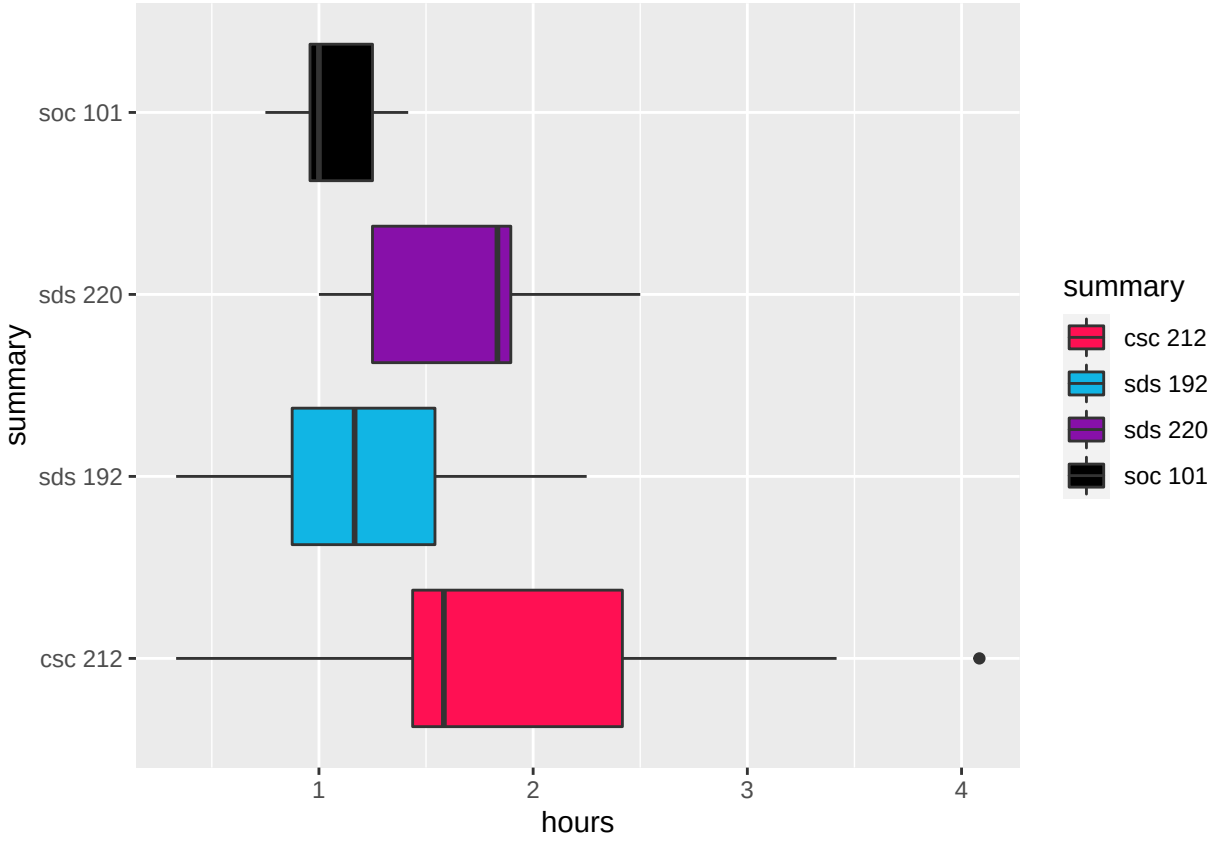
```
ggplot(calendar_data, aes(x = date, y = hours, fill = summary)) + geom_col(
  position = position_dodge(preserve = "single")) + labs(
  x = "Date",
  y = "Hours",
  title = "Comparison of Class Work Hours",
  fill = "Class"
) +
  scale_fill_manual(
  values = c("#FF1053", "#11B5E4", "#8710A9", "#000000")) +
  scale_x_date(breaks = date_breaks("day"), labels = date_format("%b %d"))
```




```
ggplot(calendar_data,
       aes(x = date, y = hours, fill = summary)) +
  geom_col(position = "fill") + scale_fill_manual(
            values = c("#FF1053", "#11B5E4", "#8710A9", "#000000"))
```



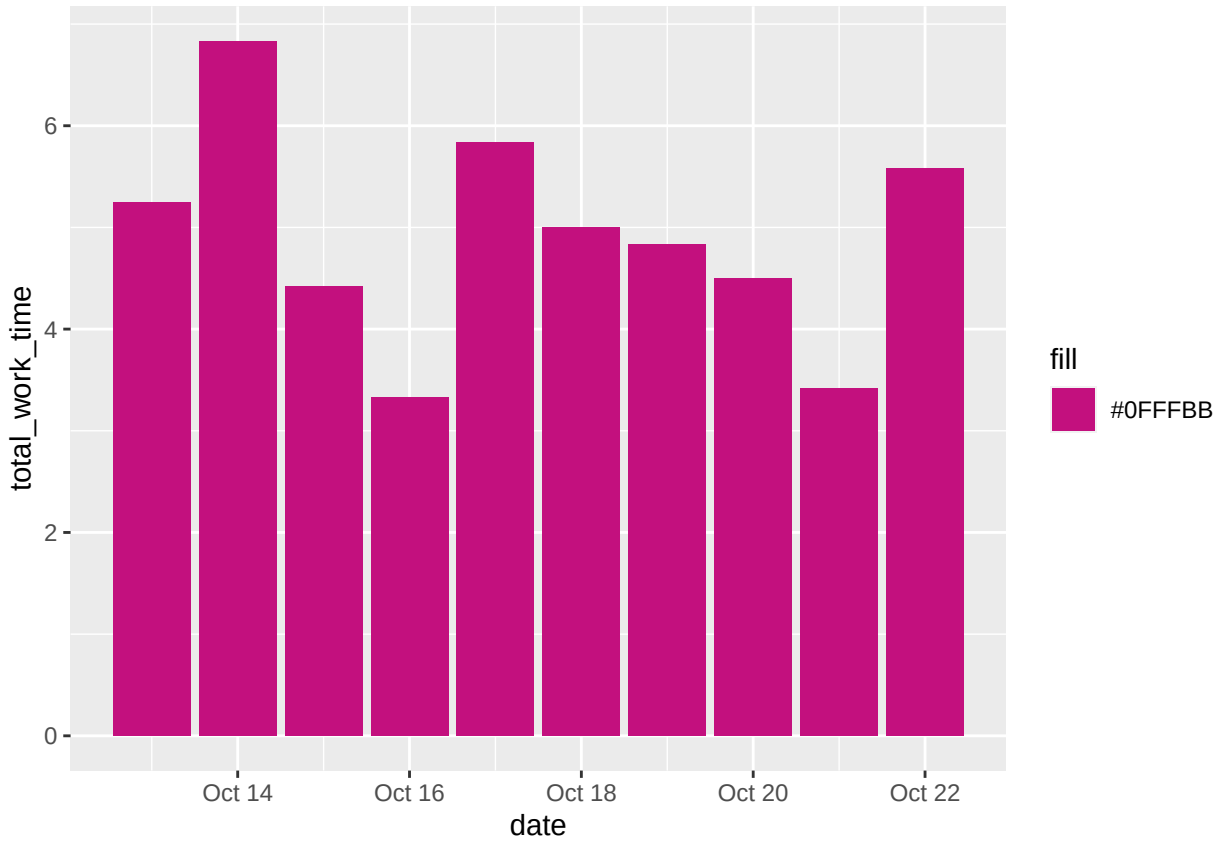

```
ggplot(calendar_data, aes(x = hours, y = summary, fill = summary)) +
  geom_boxplot() +
  scale_fill_manual(values = c("#FF1053", "#11B5E4", "#8710A9", "#000000"))
```



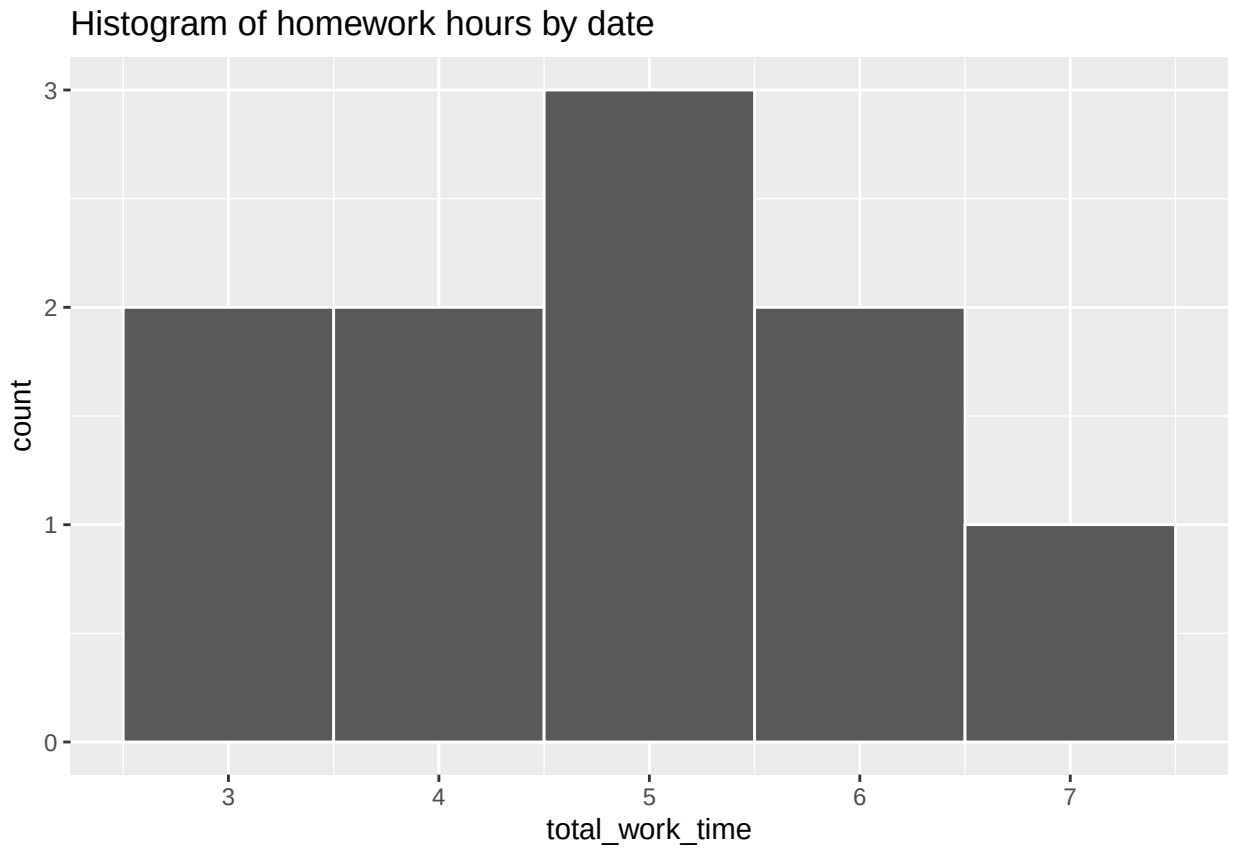

```
date_summary <- calendar_data %>%
  group_by(date) %>%
  summarize(total_work_time = sum(hours))

date_summary
```

```
## # A tibble: 10 x 2
##    date       total_work_time
##    <date>               <dbl>
##  1 2021-10-13            5.25
##  2 2021-10-14            6.83
##  3 2021-10-15            4.42
##  4 2021-10-16            3.33
##  5 2021-10-17            5.83
##  6 2021-10-18            5
##  7 2021-10-19            4.83
##  8 2021-10-20            4.5
##  9 2021-10-21            3.42
## 10 2021-10-22            5.58
```

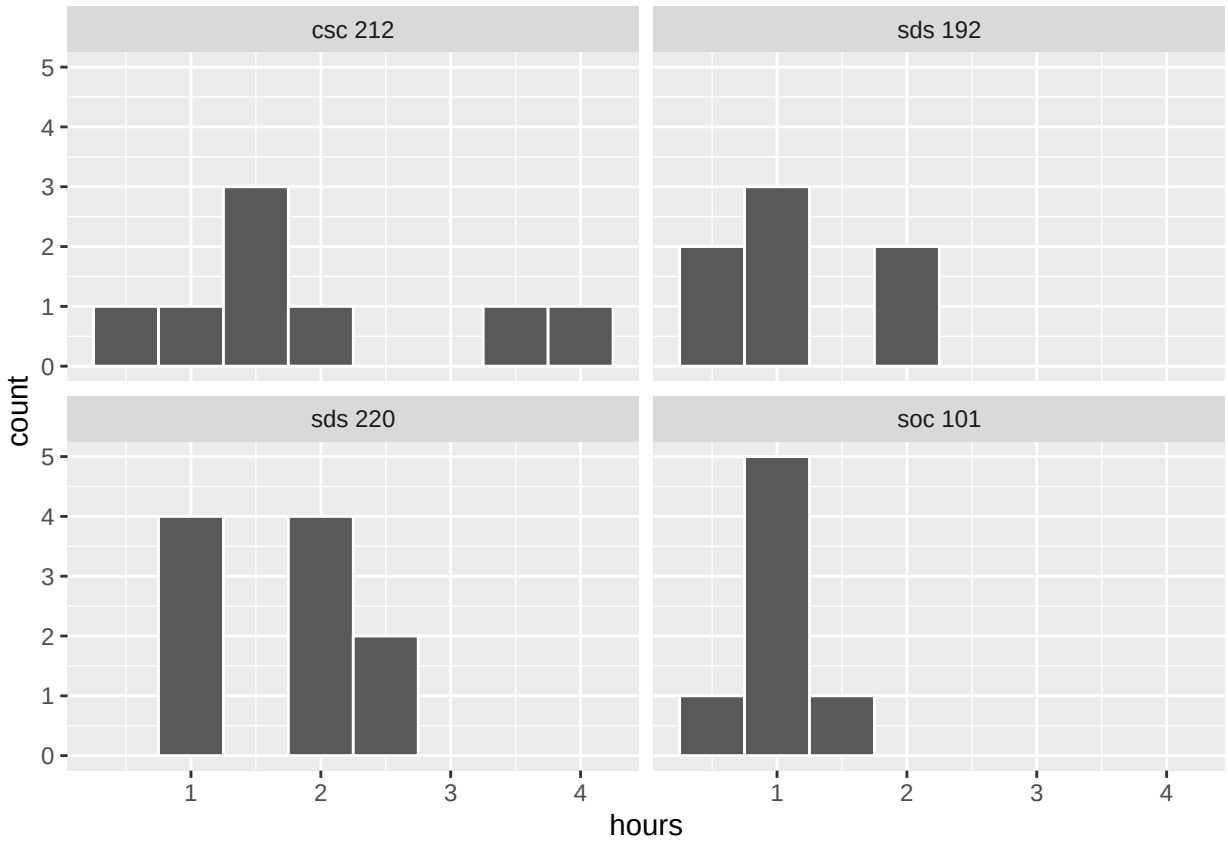
```
ggplot(date_summary, aes(x = date, y = total_work_time, fill = "#0FFFBB")) +
  geom_col() + scale_fill_manual(values = "#C3107E")
```




```
ggplot(date_summary, aes(x = total_work_time)) +
  geom_histogram(binwidth = 1, color = "white") + labs(
  title = "Histogram of homework hours by date"
)
```



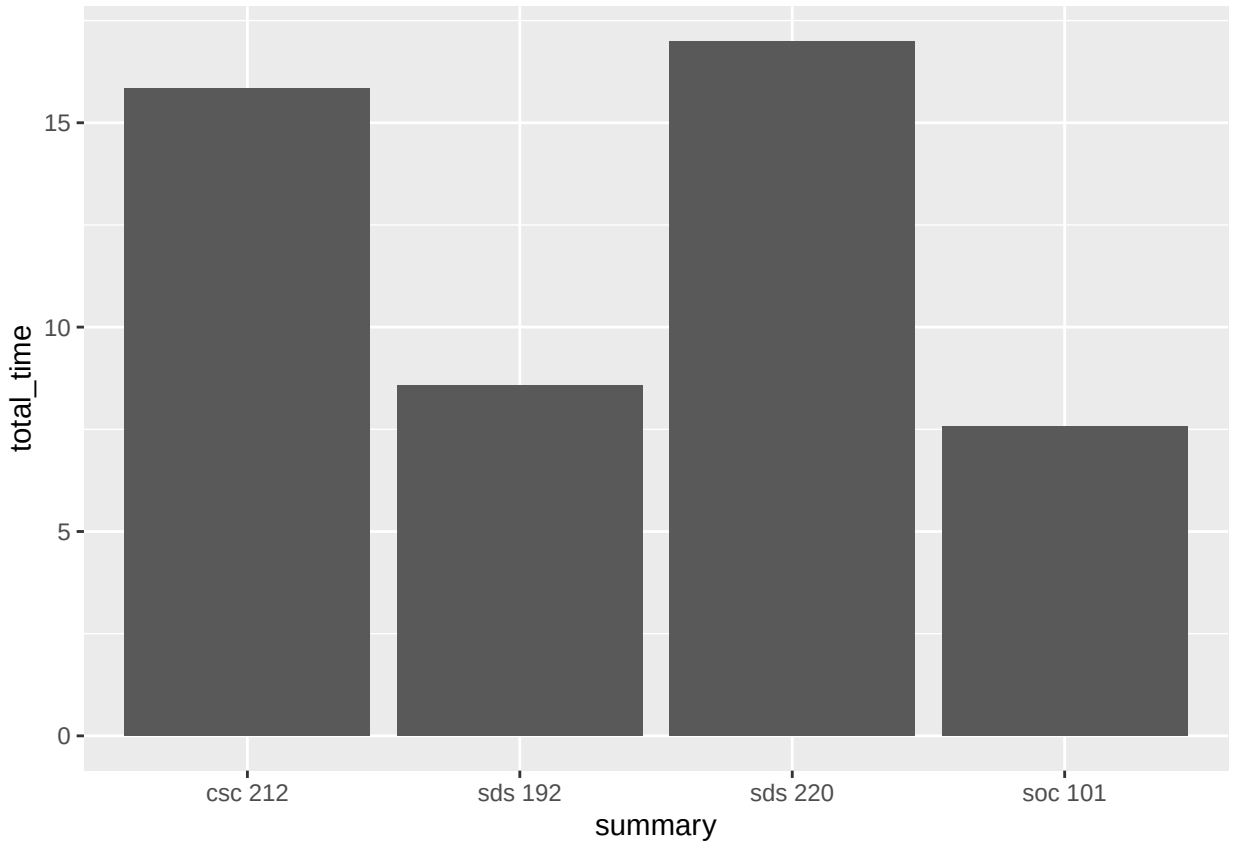

```
ggplot(calendar_data, aes(x=hours)) +
  geom_histogram(binwidth = 0.5, color = "white") + facet_wrap(~summary)
```



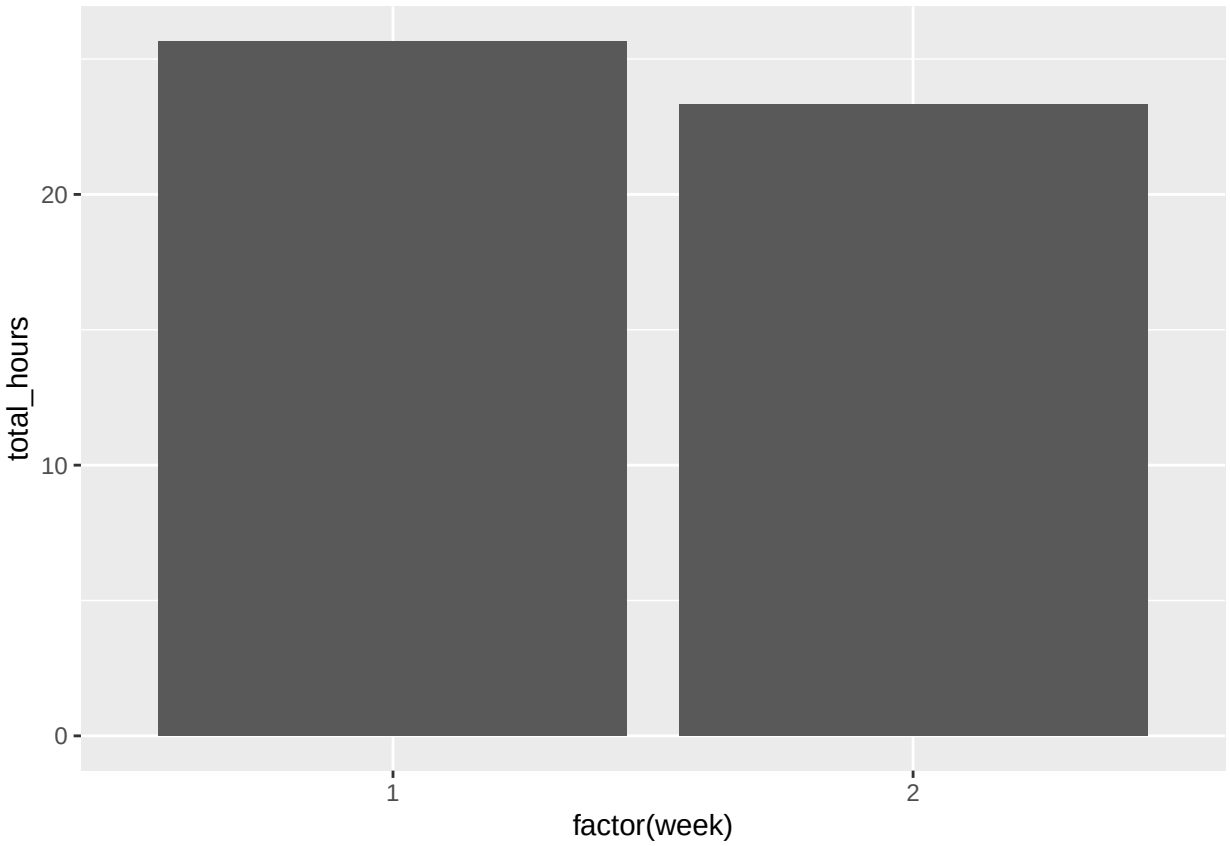

```
class_summary <- calendar_data %>%
  group_by(summary) %>%
  summarize(avg_class_time = mean(hours),
            median_class_time = median(hours),
            total_time = sum(hours)
            )
class_summary
```

```
## # A tibble: 4 x 4
##   summary avg_class_time median_class_time total_time
##   <chr>            <dbl>             <dbl>      <dbl>
## 1 csc 212           1.98              1.58      15.8
## 2 sds 192           1.23              1.17       8.58
## 3 sds 220           1.7               1.83      17
## 4 soc 101           1.08              1          7.58
```

```
ggplot(class_summary, aes(x = summary, y = total_time)) + geom_col()
```




```
calendar_data <- calendar_data %>%
  group_by(date) %>%
  arrange(date) %>%
  mutate(week = ifelse(date < "2021-10-18", 1, 2))

week_summary <- calendar_data %>%
  group_by(week) %>%
  summarize(
    total_hours = sum(hours)
  )

ggplot(week_summary,  aes(x = factor(week), y = total_hours)) + geom_col()
```


total_hours
20
10
0
1
2
factor(week)