Exercise Chapter 2

9/24/2020

What is this file here?

This is a RMarkdown file. It allows you to combine normal text with executable code - much like a Jupyter Notebook. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

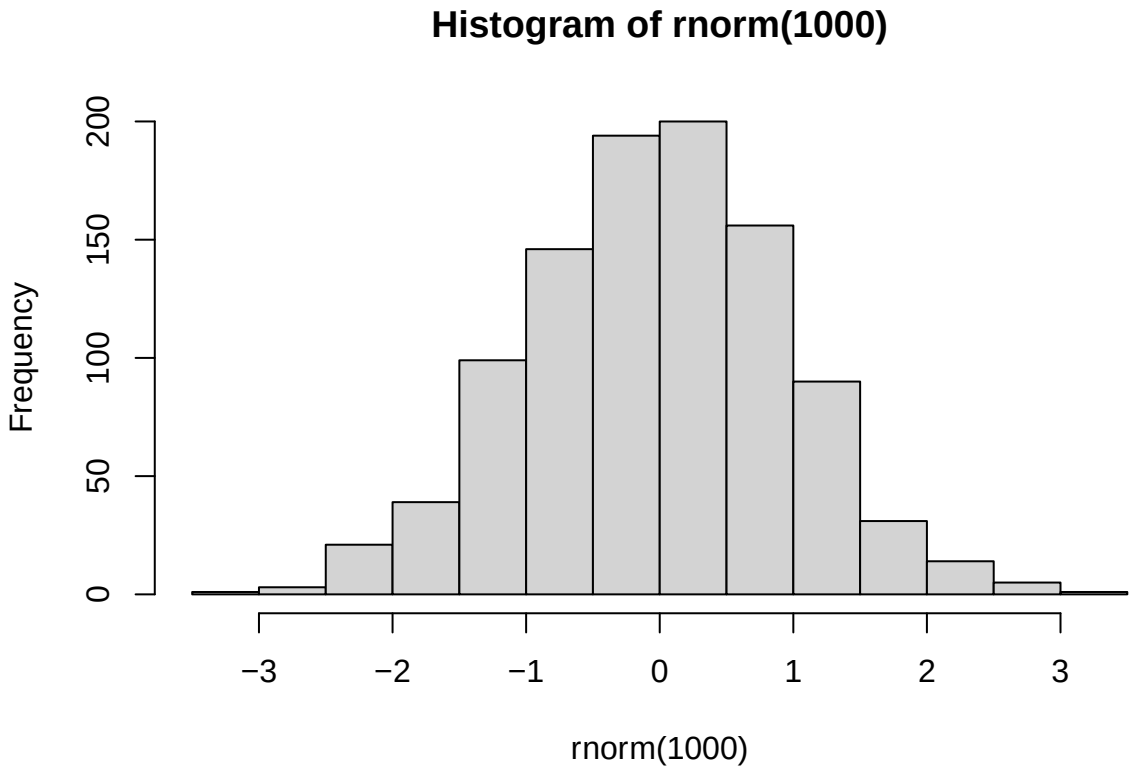
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

You can also embed plots, for example:

```
hist(rnorm(1000))
```



You can also run individual cells by putting the cursor into it and doing command + enter, or in RStudio by clicking the green "play" icon in the top-right corner of the cell. You can also run all cells of the RMarkdown file sequentially by clicking on "Run" in the top right corner of this window. This embeds all output of the cells (be it a plot or text returned to the R console).

Actual exercises

1. Outlier removal

a. Based on the half-hourly dataset for site CH-Lae, aggregated to daily means, **identify outliers** in `GPP_NT_VUT_REF` with respect to the linear relationship between `GPP_NT_VUT_REF` and `PPFD_IN`. To do so, first think about whether your data is ready to use. Then fit a linear regression model using `lm()`. This function returns a list of objects, one of which is `residuals`. Determine outliers as the "outlying" points in the distribution of residuals. See the definition of the boxplot in the bonus tutorial section of Chapter 2 and/or find the relevant information you need. You may use the base-R function `boxplot.stats()` and set the argument `coef` accordingly to our customised threshold definition.

```
# enter your solution here
## load library required
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

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

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

```
## load post-processed data in the tutorial
load("~/02_data_wrangling/data/FLX_CH-Lae_FLUXNET2015_FULLSET_HH_2004-2014_1-3_CLEAN.RData")

## view the dataset
head(hhdf)
```

```
## # A tibble: 6 x 20
##   TIMESTAMP_START     TIMESTAMP_END        TA_F SW_IN_F LW_IN_F VPD_F  PA_F
##   <dttm>              <dttm>              <dbl>   <dbl>   <dbl> <dbl> <dbl>
## 1 2004-01-01 00:00:00 2004-01-01 00:30:00    NA      NA    304.    NA  93.3
## 2 2004-01-01 00:30:00 2004-01-01 01:00:00    NA      NA    304.    NA  93.3
## 3 2004-01-01 01:00:00 2004-01-01 01:30:00    NA      NA    281.    NA  93.3
## 4 2004-01-01 01:30:00 2004-01-01 02:00:00    NA      NA    281.    NA  93.3
## 5 2004-01-01 02:00:00 2004-01-01 02:30:00    NA      NA    281.    NA  93.3
```

```
## 6 2004-01-01 02:30:00 2004-01-01 03:00:00    NA      NA    281.    NA  93.3
## # ... with 13 more variables: P_F <dbl>, WS_F <dbl>, CO2_F_MDS <dbl>,
## #   PPFD_IN <dbl>, GPP_NT_VUT_REF <dbl>, SWC_F_MDS_1 <dbl>, SWC_F_MDS_2 <dbl>,
## #   SWC_F_MDS_3 <dbl>, WS <dbl>, WD <dbl>, RH <dbl>, NIGHT <dbl>,
## #   NEE_VUT_REF_QC <dbl>
```

```
names(hhdf)
```

```
##  [1] "TIMESTAMP_START" "TIMESTAMP_END"   "TA_F"            "SW_IN_F"
##  [5] "LW_IN_F"         "VPD_F"           "PA_F"            "P_F"
##  [9] "WS_F"            "CO2_F_MDS"       "PPFD_IN"         "GPP_NT_VUT_REF"
## [13] "SWC_F_MDS_1"     "SWC_F_MDS_2"     "SWC_F_MDS_3"     "WS"
## [17] "WD"              "RH"              "NIGHT"           "NEE_VUT_REF_QC"
```

```
summary(hhdf)
```

```
##  TIMESTAMP_START               TIMESTAMP_END                     TA_F
##  Min.   :2004-01-01 00:00:00   Min.   :2004-01-01 00:30:00   Min.   :-17.200
##  1st Qu.:2006-10-01 11:52:30   1st Qu.:2006-10-01 12:22:30   1st Qu.:  1.386
##  Median :2009-07-01 23:45:00   Median :2009-07-02 00:15:00   Median :  7.840
##  Mean   :2009-07-01 23:45:00   Mean   :2009-07-02 00:15:00   Mean   :  7.679
##  3rd Qu.:2012-04-01 11:37:30   3rd Qu.:2012-04-01 12:07:30   3rd Qu.: 13.740
##  Max.   :2014-12-31 23:30:00   Max.   :2015-01-01 00:00:00   Max.   : 31.820
##                                                              NA's   :13449
##     SW_IN_F            LW_IN_F          VPD_F             PA_F
##  Min.   :   0.000   Min.   :135.4   Min.   : 0.000   Min.   :89.57
##  1st Qu.:   0.000   1st Qu.:275.1   1st Qu.: 0.179   1st Qu.:92.86
##  Median :   2.339   Median :310.4   Median : 1.640   Median :93.34
##  Mean   : 135.961   Mean   :304.3   Mean   : 3.234   Mean   :93.26
##  3rd Qu.: 175.530   3rd Qu.:337.4   3rd Qu.: 4.734   3rd Qu.:93.74
##  Max.   :1074.410   Max.   :423.9   Max.   :34.391   Max.   :95.32
##  NA's   :13635                      NA's   :13449
##       P_F               WS_F          CO2_F_MDS        PPFD_IN
##  Min.   :0.00000   Min.   : 0.004   Min.   :209.0   Min.   :   3.399
##  1st Qu.:0.00000   1st Qu.: 1.146   1st Qu.:370.3   1st Qu.:   4.119
##  Median :0.00000   Median : 2.027   Median :386.3   Median :   9.860
##  Mean   :0.06702   Mean   : 2.463   Mean   :395.7   Mean   : 284.518
##  3rd Qu.:0.06000   3rd Qu.: 3.328   3rd Qu.:401.7   3rd Qu.: 355.300
##  Max.   :3.55100   Max.   :13.249   Max.   :999.4   Max.   :2170.000
##                    NA's   :9652     NA's   :6023    NA's   :13778
##  GPP_NT_VUT_REF     SWC_F_MDS_1     SWC_F_MDS_2      SWC_F_MDS_3
##  Min.   :-35.469   Min.   : 7.70   Min.   : 6.505   Min.   : 7.32
##  1st Qu.: -0.390   1st Qu.:18.44   1st Qu.:17.271   1st Qu.:18.55
##  Median :  1.660   Median :21.06   Median :21.810   Median :21.43
##  Mean   :  4.868   Mean   :21.38   Mean   :21.106   Mean   :21.17
##  3rd Qu.:  7.747   3rd Qu.:24.79   3rd Qu.:24.560   3rd Qu.:23.89
##  Max.   : 61.175   Max.   :32.85   Max.   :32.086   Max.   :31.79
##  NA's   :16732     NA's   :32047   NA's   :21211    NA's   :21264
##        WS               WD               RH              NIGHT
##  Min.   : 0.004   Min.   :  0.062   Min.   : 17.09   Min.   :0.0000
##  1st Qu.: 1.146   1st Qu.:101.196   1st Qu.: 65.76   1st Qu.:0.0000
##  Median : 2.027   Median :226.910   Median : 82.00   Median :0.0000
##  Mean   : 2.463   Mean   :187.940   Mean   : 78.95   Mean   :0.4776
##  3rd Qu.: 3.328   3rd Qu.:259.792   3rd Qu.: 97.10   3rd Qu.:1.0000
##  Max.   :13.249   Max.   :359.987   Max.   :100.00   Max.   :1.0000
```

```
##  NA's   :9652     NA's   :8552      NA's   :14591
##  NEE_VUT_REF_QC
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :1.0000
##  Mean   :0.8328
##  3rd Qu.:1.0000
##  Max.   :3.0000
##
```

```
## aggregated to daily means and only keep the GPP and PPFD_IN that we are interested
ddf <- hhdf %>%
  mutate(date = as_date(TIMESTAMP_START)) %>%  # converts the ymd_hm-formatted date-time object to a da
  group_by(date) %>%
  summarise(GPP_NT_VUT_REF = mean(GPP_NT_VUT_REF, na.rm = TRUE),
            PPFD_IN = mean(PPFD_IN, na.rm = TRUE)
            )
head(ddf)
```

```
## # A tibble: 6 x 3
##   date       GPP_NT_VUT_REF PPFD_IN
##   <date>              <dbl>   <dbl>
## 1 2004-01-01            NaN     NaN
## 2 2004-01-02            NaN     NaN
## 3 2004-01-03            NaN     NaN
## 4 2004-01-04            NaN     NaN
## 5 2004-01-05            NaN     NaN
## 6 2004-01-06            NaN     NaN
```

```
nrow(ddf)
```

```
## [1] 4018
```

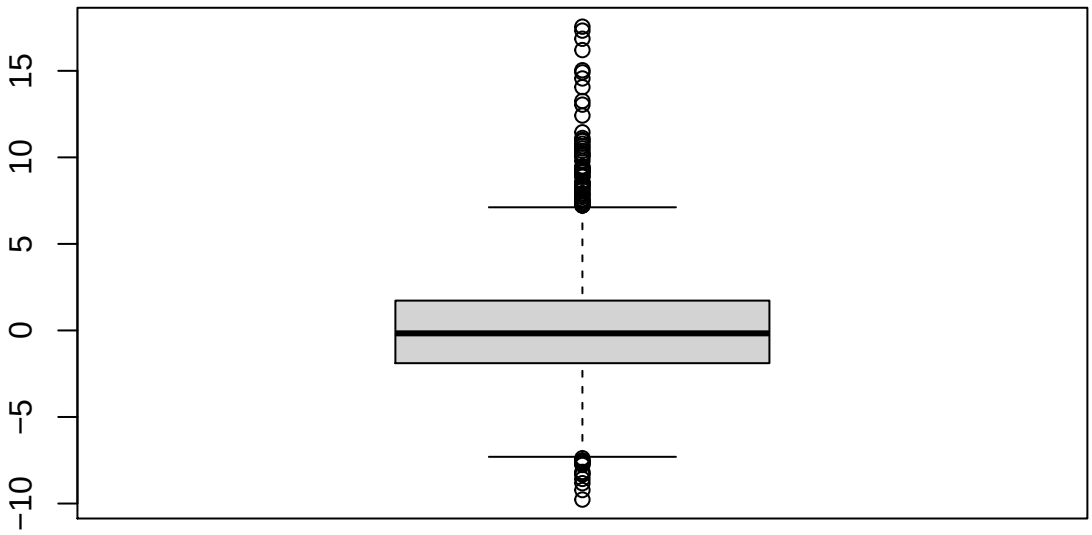
```
summary(ddf)
```

```
##       date            GPP_NT_VUT_REF      PPFD_IN
##  Min.   :2004-01-01   Min.   :-6.440   Min.   :  3.941
##  1st Qu.:2006-10-01   1st Qu.: 1.615   1st Qu.: 98.476
##  Median :2009-07-01   Median : 4.124   Median :228.992
##  Mean   :2009-07-01   Mean   : 5.019   Mean   :284.557
##  3rd Qu.:2012-03-31   3rd Qu.: 8.404   3rd Qu.:456.231
##  Max.   :2014-12-31   Max.   :34.035   Max.   :790.339
##                       NA's   :106      NA's   :276
```

```
## do the linear regression for GPP_NT_VUT_REF with respect to PPFD_IN
GPP_lm <- lm(GPP_NT_VUT_REF~PPFD_IN, data = ddf)
GPP_lm_res <- GPP_lm$residuals
summary(GPP_lm_res)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -9.7721 -1.8921 -0.1674  0.0000  1.7239 17.5503
```

```
## boxplot the residual to find outliers
boxplot(GPP_lm_res)
```




```
box_coef <- 1.5
GPP_lm_res_boxstat <- boxplot.stats(GPP_lm_res, coef = box_coef, do.conf = F, do.out = T)
cat('ratio of outliers: ',length(GPP_lm_res_boxstat$out)/GPP_lm_res_boxstat$n, 'under coef = ',box_coef)
```

```
## ratio of outliers:  0.02197802 under coef =  1.5
```

```
cat('range of remaining data: ', GPP_lm_res_boxstat$stats[1], 'to ',GPP_lm_res_boxstat$stats[5])
```

```
## range of remaining data:  -7.300733 to  7.114709
```

```
GPP_lm$residuals[1:10]
```

```
##         264         265         266         267         268         269
## -0.27743024  1.13651894  1.95341127  0.28860552  0.77972354  0.09048646
##         270         271         272         273
## -0.50580476  1.35083541 -0.78872659 -0.31363420
```

```
GPP_lm$fitted.values[1:10]
```

```
##      264      265      266      267      268      269      270      271
## 7.349846 3.306800 3.716173 2.440840 4.276417 1.585508 1.562366 2.895286
##      272      273
## 3.287955 2.724735
```

```
ddf$GPP_NT_VUT_REF %>% length()
```

```
## [1] 4018
```

```
GPP_lm$fitted.values %>% length()
```

```
## [1] 3731
```

```
## find out the outlier positions
Outliers_position <- GPP_lm_res_boxstat$out %>% names() %>% as.numeric()

## Just to confirm outlier position is correct
(ddf[Outliers_position,]$GPP_NT_VUT_REF[1] - GPP_lm$fitted.values[1] + GPP_lm$residuals[1]) < 1e-5
```

```
##  264
## TRUE
```

b. **Remove outliers** by setting values in the data frame (aggregated daily data frame for CH-Lae) to

NA.

In base-R, this could be done (admittedly quite simply) as:

```
# enter your solution here
## The position need to be removed
ddf_1 <- ddf
ddf_1$outlier_flag <- 0
ddf_1$outlier_flag[Outliers_position] <- 1
ddf_1$GPP_NT_VUT_REF[Outliers_position] <- NA
ddf_1$PPFD_IN[Outliers_position] <- NA
head(ddf_1[Outliers_position,])
```

```
## # A tibble: 6 x 4
##   date       GPP_NT_VUT_REF PPFD_IN outlier_flag
##   <date>              <dbl>   <dbl>        <dbl>
## 1 2005-04-29             NA      NA            1
## 2 2005-04-30             NA      NA            1
## 3 2005-05-21             NA      NA            1
## 4 2005-10-18             NA      NA            1
## 5 2006-01-14             NA      NA            1
## 6 2006-01-15             NA      NA            1
```

With dplyr:

```
# enter your solution here
ddf_2 <- ddf %>% mutate(outlier_flag = 0)
ddf_2$outlier_flag[Outliers_position] <- 1
ddf_2 %>% filter(outlier_flag == 1) %>% head()
```

```
## # A tibble: 6 x 4
##   date       GPP_NT_VUT_REF PPFD_IN outlier_flag
##   <date>              <dbl>   <dbl>        <dbl>
## 1 2005-04-29          0.414   525.             1
## 2 2005-04-30          1.35    631.             1
## 3 2005-05-21         13.0     262.             1
## 4 2005-10-18         11.0      82.7            1
## 5 2006-01-14         14.6      54.3            1
## 6 2006-01-15         10.9     121.             1
```

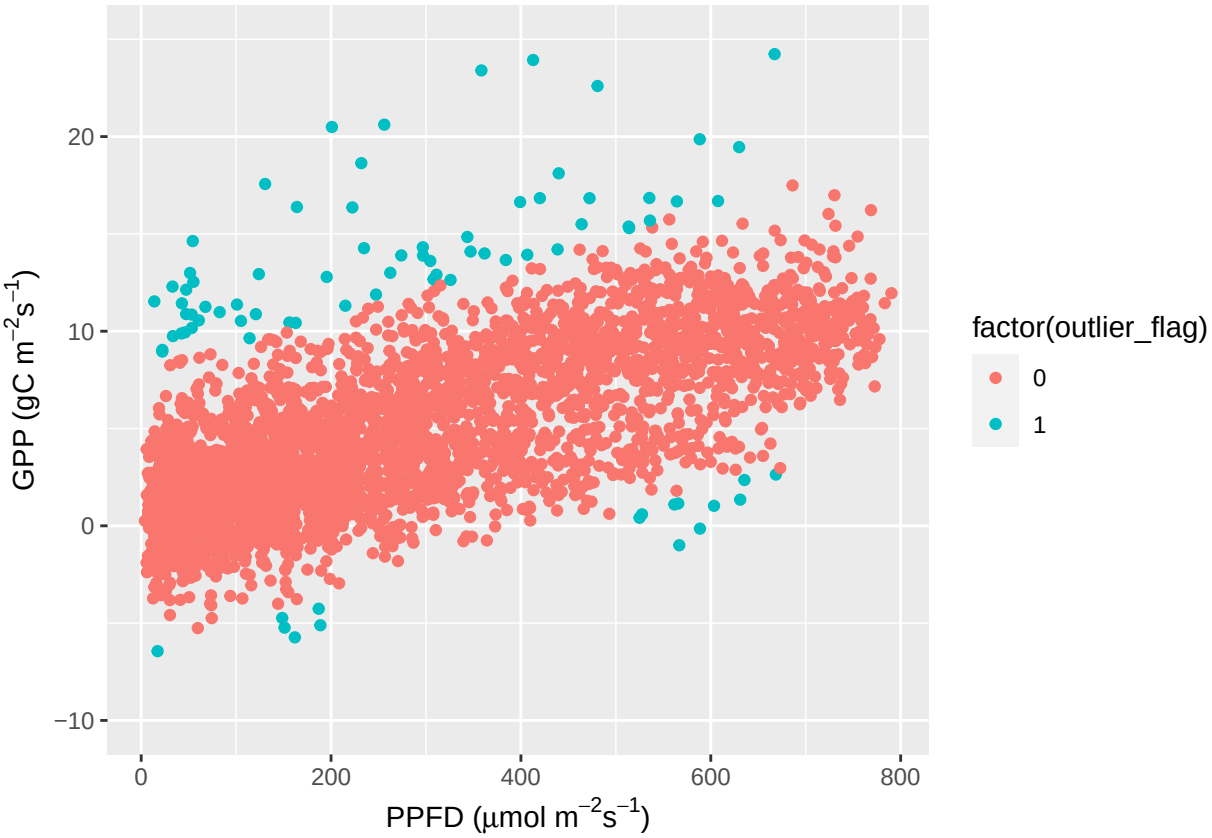
```
ddf_3 <- ddf_2 %>%
  mutate(GPP_NT_VUT_REF = ifelse(outlier_flag == 0, GPP_NT_VUT_REF, NA),
         PPFD_IN = ifelse(outlier_flag == 0, PPFD_IN, NA))

head(ddf_3[Outliers_position,])
```

```
## # A tibble: 6 x 4
##   date       GPP_NT_VUT_REF PPFD_IN outlier_flag
##   <date>              <dbl>   <dbl>        <dbl>
## 1 2005-04-29             NA      NA            1
## 2 2005-04-30             NA      NA            1
## 3 2005-05-21             NA      NA            1
## 4 2005-10-18             NA      NA            1
## 5 2006-01-14             NA      NA            1
## 6 2006-01-15             NA      NA            1
```

c. Create a scatterplot of all daily data (GPP vs. PPFD) and highlight outliers that are removed by step b.

```
# enter your solution here
## use ddf_2 to do the plotting here
ddf_2 %>% ggplot(aes(x=PPFD_IN,y=GPP_NT_VUT_REF,color=factor(outlier_flag))) +
  geom_point()+
  labs(x = expression(paste("PPFD (", mu, "mol m"^-2, "s"^-1, ")")), y = expression(paste("GPP (gC m"^-2
  ylim(-10, 25)
```

```
## Warning: Removed 287 rows containing missing values (geom_point).
```



2. Visualising diurnal and seasonal cycles

Using the half-hourly dataset for site CH-Lae, visualise how GPP (`GPP_NT_VUT_REF`) varies on two time scales: diurnal (within-day at hourly time scale) and seasonal. To implement this, follow the following steps:

a. Summarise half-hourly data for each data across multiple years to get a mean seasonality with a mean diurnal cycle for each day of the year. You will use functions from the lubridate package (e.g., `yday()`). To deal with date-time objects, use the lubridate package. Enter `?day` to get more hints.

```
# enter your solution here
## load library required
library(tidyverse)
library(lubridate)

## load post-processed data in the tutorial
load("~/02_data_wrangling/data/FLX_CH-Lae_FLUXNET2015_FULLSET_HH_2004-2014_1-3_CLEAN.RData")

## view the dataset
head(hhdf)
```

```
## # A tibble: 6 x 20
##   TIMESTAMP_START     TIMESTAMP_END        TA_F SW_IN_F LW_IN_F VPD_F  PA_F
##   <dttm>              <dttm>              <dbl>   <dbl>   <dbl> <dbl> <dbl>
## 1 2004-01-01 00:00:00 2004-01-01 00:30:00    NA      NA    304.    NA  93.3
## 2 2004-01-01 00:30:00 2004-01-01 01:00:00    NA      NA    304.    NA  93.3
## 3 2004-01-01 01:00:00 2004-01-01 01:30:00    NA      NA    281.    NA  93.3
## 4 2004-01-01 01:30:00 2004-01-01 02:00:00    NA      NA    281.    NA  93.3
## 5 2004-01-01 02:00:00 2004-01-01 02:30:00    NA      NA    281.    NA  93.3
## 6 2004-01-01 02:30:00 2004-01-01 03:00:00    NA      NA    281.    NA  93.3
## # ... with 13 more variables: P_F <dbl>, WS_F <dbl>, CO2_F_MDS <dbl>,
## #   PPFD_IN <dbl>, GPP_NT_VUT_REF <dbl>, SWC_F_MDS_1 <dbl>, SWC_F_MDS_2 <dbl>,
## #   SWC_F_MDS_3 <dbl>, WS <dbl>, WD <dbl>, RH <dbl>, NIGHT <dbl>,
## #   NEE_VUT_REF_QC <dbl>
```

```
names(hhdf)
```

```
##  [1] "TIMESTAMP_START" "TIMESTAMP_END"   "TA_F"            "SW_IN_F"
##  [5] "LW_IN_F"         "VPD_F"           "PA_F"            "P_F"
##  [9] "WS_F"            "CO2_F_MDS"       "PPFD_IN"         "GPP_NT_VUT_REF"
## [13] "SWC_F_MDS_1"     "SWC_F_MDS_2"     "SWC_F_MDS_3"     "WS"
## [17] "WD"              "RH"              "NIGHT"           "NEE_VUT_REF_QC"
```

```
summary(hhdf)
```

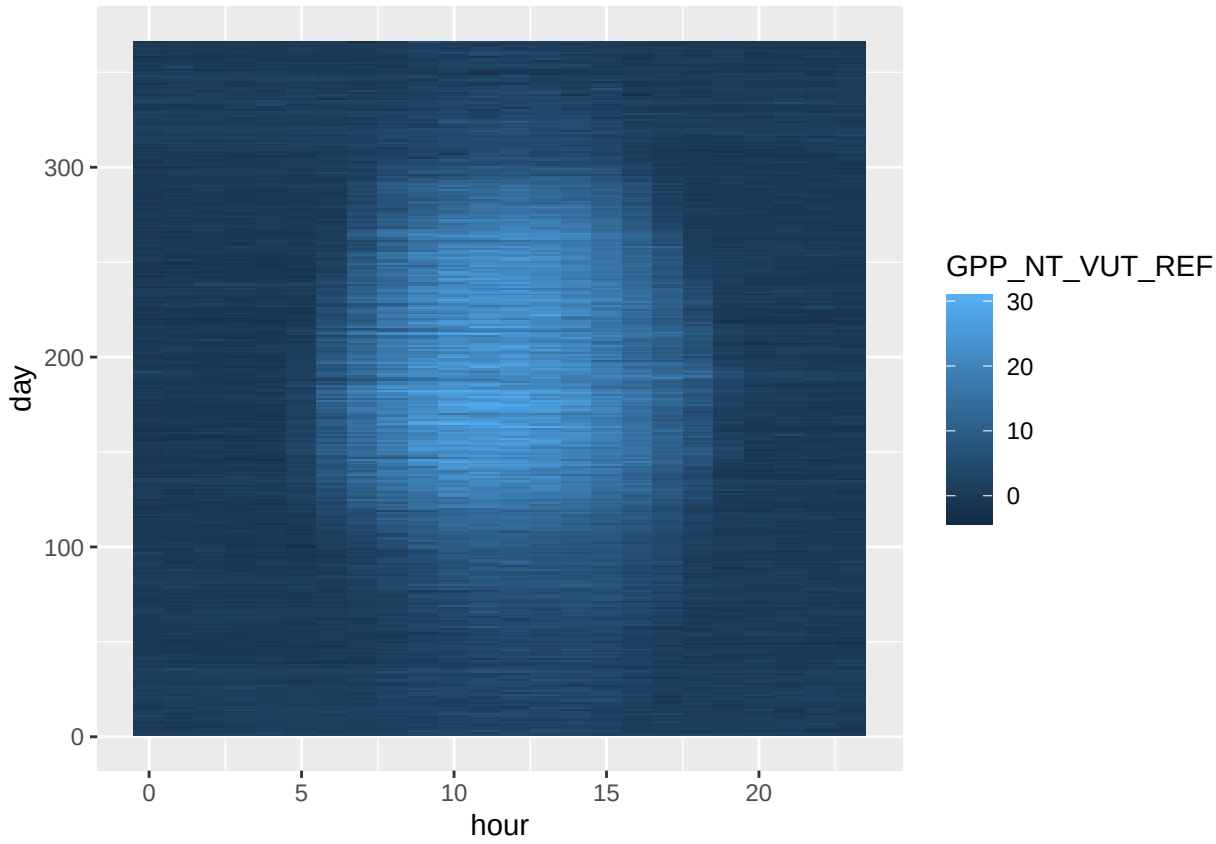
```
##  TIMESTAMP_START               TIMESTAMP_END                     TA_F
##  Min.   :2004-01-01 00:00:00   Min.   :2004-01-01 00:30:00   Min.   :-17.200
##  1st Qu.:2006-10-01 11:52:30   1st Qu.:2006-10-01 12:22:30   1st Qu.:  1.386
##  Median :2009-07-01 23:45:00   Median :2009-07-02 00:15:00   Median :  7.840
##  Mean   :2009-07-01 23:45:00   Mean   :2009-07-02 00:15:00   Mean   :  7.679
##  3rd Qu.:2012-04-01 11:37:30   3rd Qu.:2012-04-01 12:07:30   3rd Qu.: 13.740
##  Max.   :2014-12-31 23:30:00   Max.   :2015-01-01 00:00:00   Max.   : 31.820
##                                                              NA's   :13449
##     SW_IN_F            LW_IN_F          VPD_F             PA_F
##  Min.   :   0.000   Min.   :135.4   Min.   : 0.000   Min.   :89.57
##  1st Qu.:   0.000   1st Qu.:275.1   1st Qu.: 0.179   1st Qu.:92.86
##  Median :   2.339   Median :310.4   Median : 1.640   Median :93.34
##  Mean   : 135.961   Mean   :304.3   Mean   : 3.234   Mean   :93.26
##  3rd Qu.: 175.530   3rd Qu.:337.4   3rd Qu.: 4.734   3rd Qu.:93.74
##  Max.   :1074.410   Max.   :423.9   Max.   :34.391   Max.   :95.32
##  NA's   :13635                      NA's   :13449
##       P_F               WS_F          CO2_F_MDS        PPFD_IN
##  Min.   :0.00000   Min.   : 0.004   Min.   :209.0   Min.   :   3.399
##  1st Qu.:0.00000   1st Qu.: 1.146   1st Qu.:370.3   1st Qu.:   4.119
##  Median :0.00000   Median : 2.027   Median :386.3   Median :   9.860
##  Mean   :0.06702   Mean   : 2.463   Mean   :395.7   Mean   : 284.518
##  3rd Qu.:0.06000   3rd Qu.: 3.328   3rd Qu.:401.7   3rd Qu.: 355.300
##  Max.   :3.55100   Max.   :13.249   Max.   :999.4   Max.   :2170.000
##                    NA's   :9652     NA's   :6023    NA's   :13778
##  GPP_NT_VUT_REF     SWC_F_MDS_1     SWC_F_MDS_2      SWC_F_MDS_3
##  Min.   :-35.469   Min.   : 7.70   Min.   : 6.505   Min.   : 7.32
##  1st Qu.: -0.390   1st Qu.:18.44   1st Qu.:17.271   1st Qu.:18.55
##  Median :  1.660   Median :21.06   Median :21.810   Median :21.43
##  Mean   :  4.868   Mean   :21.38   Mean   :21.106   Mean   :21.17
##  3rd Qu.:  7.747   3rd Qu.:24.79   3rd Qu.:24.560   3rd Qu.:23.89
##  Max.   : 61.175   Max.   :32.85   Max.   :32.086   Max.   :31.79
```

```
##  NA's   :16732     NA's   :32047   NA's   :21211    NA's   :21264
##        WS               WD                RH             NIGHT
##  Min.   : 0.004   Min.   :  0.062   Min.   : 17.09   Min.   :0.0000
##  1st Qu.: 1.146   1st Qu.:101.196   1st Qu.: 65.76   1st Qu.:0.0000
##  Median : 2.027   Median :226.910   Median : 82.00   Median :0.0000
##  Mean   : 2.463   Mean   :187.940   Mean   : 78.95   Mean   :0.4776
##  3rd Qu.: 3.328   3rd Qu.:259.792   3rd Qu.: 97.10   3rd Qu.:1.0000
##  Max.   :13.249   Max.   :359.987   Max.   :100.00   Max.   :1.0000
##  NA's   :9652     NA's   :8552      NA's   :14591
##  NEE_VUT_REF_QC
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :1.0000
##  Mean   :0.8328
##  3rd Qu.:1.0000
##  Max.   :3.0000
##
```

```
yday_hour_df <- hhdf %>%
  mutate(day = yday(TIMESTAMP_START), hour = hour(TIMESTAMP_START)) %>%  # converts the ymd_hm-formatte
  group_by(day, hour) %>%
  summarise(GPP_NT_VUT_REF = mean(GPP_NT_VUT_REF, na.rm = TRUE)
            )
```

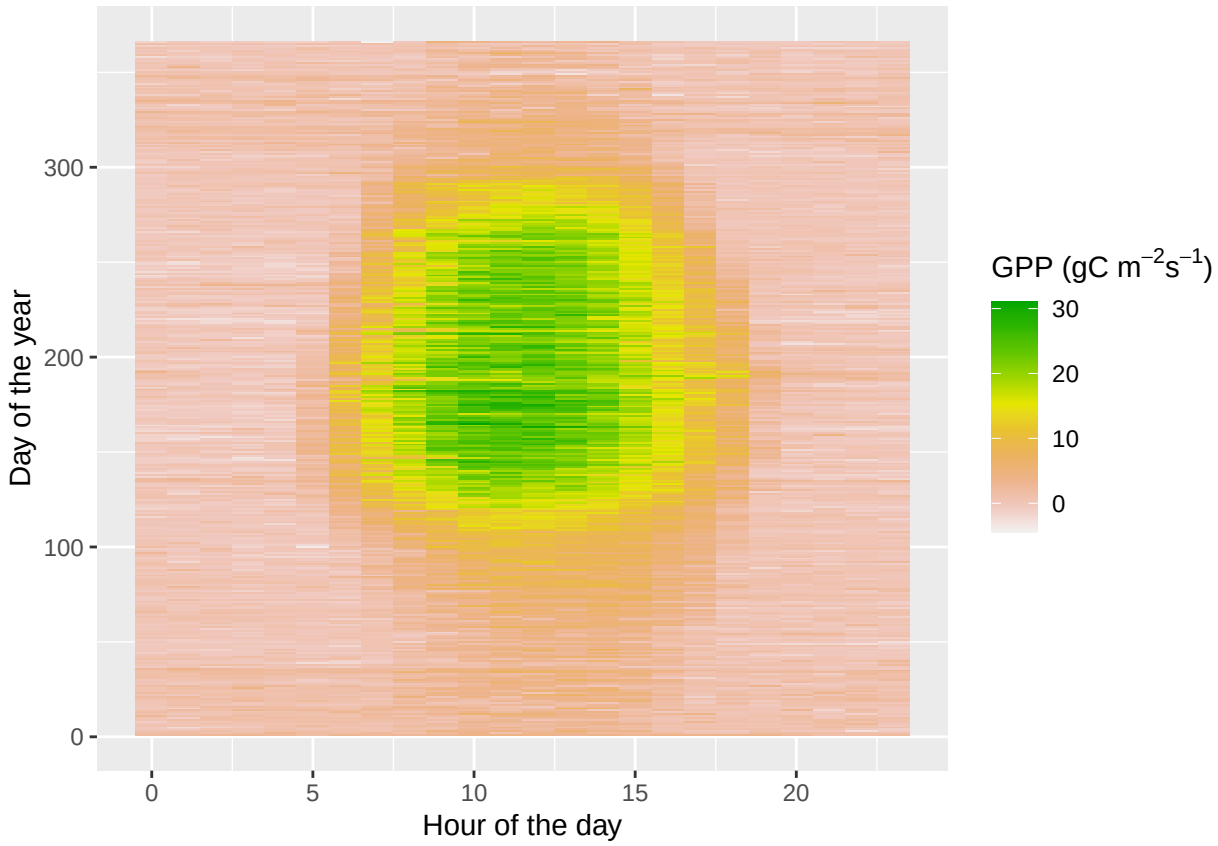
b. Create a raster plot (`geom_raster()`), mapping the hour of the day to the x-axis, the day of the year to the y-axis, and the magnitude of `GPP_NT_VUT_REF` to color (fill).

```
# enter your solution here
gg <- yday_hour_df %>% ggplot(aes(x = hour, y = day)) +
  geom_raster(aes(fill = GPP_NT_VUT_REF))
print(gg)
```



c. Make this figure ready for publication by adding nice labels and choosing a good color scale.

```
# enter your solution here
gg + labs(x = 'Hour of the day', y = 'Day of the year', fill = expression(paste("GPP (gC m"^-2, "s"^-1,
  scale_fill_gradientn(colours = rev(terrain.colors(10)))
```


Day of the year
300
200
100
0
0
5
10
15
20
Hour of the day
GPP (gC m^{-2}s^{-1})
30
20
10
0