MOF-801 Adsorption Isotherm of Water

T.L. Hsieh

2023-01-18

### view the dataset with table and plot

dataset <- read\_excel("dataset.xlsx")  
colnames(dataset)

## [1] "time\_s" "hpv" "uptake"

head(dataset)

## # A tibble: 6 × 3  
## time\_s hpv uptake  
## <dbl> <dbl> <dbl>  
## 1 0 68.9 0.0587  
## 2 30 88.9 0.0742  
## 3 60 80.0 NA   
## 4 90 73.3 0.108   
## 5 120 69.4 0.120   
## 6 150 66.3 0.132

p <- ggplot(data=dataset)+  
 geom\_point(mapping=aes(x=time\_s/3600, y=uptake\*100,,color='uptake'))+  
 geom\_point(mapping=aes(x=time\_s/3600, y=hpv, color='humidity'))+  
 ylab("water uptake/ humidity")+  
 xlab('time (hr)')+  
 theme\_test()  
  
ggplotly(p)

## PhantomJS not found. You can install it with webshot::install\_phantomjs(). If it is installed, please make sure the phantomjs executable can be found via the PATH variable.

### Convert excel file to csv

df\_excel <- read\_excel("dataset.xlsx")  
write.csv(df\_excel, "dataset.csv")

Sometimes, R is having problem with .xlsx file when plotting, so it is better to convert xlsx to csv file, which will save you among of time in a long run.

### Clean Data: reomve unwanted data points

df\_csv <- read.csv("dataset.csv")  
colnames(df\_csv)

## [1] "X" "time\_s" "hpv" "uptake"

# data clean  
df\_clean <- df\_csv %>% filter(time\_s/3600>=3) %>%   
 filter(!((time\_s/3600 >= 5.267 & time\_s/3600 <= 5.359) & (hpv >= 14.80 & hpv <= 20.28)))

Here we use pipe operator (%>%) to clean the data. First, we use filter function to extract the data with ‘time\_s/3600’ greater than 3, which means only the data collected after 3 hours of measurement will be included. Second, we remove the points with the time within 5.267 to 5.359 hours and the hpv (humidity) between 14.80 to 20.28 because it will affect the accuracy of the finally result. Finally, we checked the cleaned data by visualizing it.

p1 <- ggplot(data=df\_clean)+  
 geom\_point(mapping=aes(x=time\_s/3600, y=uptake\*100, color='uptake'))+  
 geom\_point(mapping=aes(x=time\_s/3600, y=hpv, color='hpv'))+  
 ylim(0, 100)+  
 labs(x="time (min)", y="uptake/ humidity")+  
 theme\_bw()  
  
ggplotly(p1)

### plot: water uptake vs humidity

p2 <- ggplot(data=df\_clean)+  
 geom\_point(mapping=aes(x=hpv, y=uptake\*100, color='uptake'))+  
 labs(x="Humidity (%RH)", y="water uptake (wt.%)")+  
 theme\_bw()  
  
ggplotly(p2)

With the plot above we can easily type down the maximum water uptake of the test material at certain humidity. For example, the maximum water uptake at humidity around 10 %RH will be 24.653 wt.%. After collecting all the maximum uptake at the humidity we want to include, we can save the data set as .csv file and make the adsorption isotherm plot (shown below).

humidity = c(9.50, 19.62, 29.95, 40.18, 50.33,63.30,70.02,80.05,89.91)  
water\_uptake = c(24.653,30.923, 32.788, 33.576, 34.211,34.865,35.230,35.903,37.615)  
  
point\_pick = data.frame(humidity, water\_uptake)  
write.csv(point\_pick, 'point\_pick.csv', row.names = FALSE)  
  
  
p3 <- ggplot(data=point\_pick, mapping=aes(x=humidity, y=water\_uptake, color='uptake'))+  
 geom\_point()+  
 geom\_smooth(se=FALSE)+  
 labs(x="humidity (%RH)", y="uptake (wt.%)")+  
 xlim(0,100)+  
 theme\_bw()  
  
ggplotly(p3)

## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'

### Group the data by humidity

Another way to process the data is to group the data by humidity. First we use mutate function to add extra column named “hpv\_1” with data that round of hpv to first decimal. Second we group the whole data by hpv\_1 and use summarise function to calculate the maximum and mean value of the water uptake.

df\_group <- df\_clean %>%   
 mutate(hpv\_1 = round(hpv,1)) %>%   
 group\_by(hpv\_1) %>%   
 summarise(uptake\_max = max(uptake), uptake\_mean = mean(uptake))

### Plot with smooth line

Finally, we plot the df\_group data by hpv\_1 and uptake, and add smooth line to the plot (as shown below).

p <- ggplot(data=df\_group)+  
 geom\_point(mapping=aes(x=hpv\_1, y=uptake\_mean\*100))+  
 geom\_smooth(mapping=aes(x=hpv\_1, y=uptake\_mean\*100))+  
 ylab('water uptake (wt.%)')+  
 xlab('humidity (%RH)')+  
 xlim(0,100)+  
 theme\_test()  
  
ggplotly(p)

## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'

## Warning: Removed 155 rows containing non-finite values (`stat\_smooth()`).

The figure above shows that the material adsorbs water as the increase of humidity.

### save final result

df\_dropna <- drop\_na(df\_group)  
write.csv(df\_group, "result\_r.csv", row.names = FALSE)