

```
/* After joining the weather data to the hour grouped bikes data this code was used to
   calculate the total activity per hour and per station. It was also used to find
   the frequency of station occurrences in the data. The total station activity and
   frequency were used to get a weighting per station. The total bikes in the system
   was also calculated for each hour. */

/* Connect to the CAS Server - to access the files in the public folder */
cas mySession2 host="localhost" port=5570 sessopts=(caslib=casuser timeout=18000);
caslib _all_ assign;

/* Increase system data limit to prevent restrictions */
options CASDATALIMIT=ALL;

/* Duplicating and sorting the hour grouped weather data by address and then by update time */
proc sort data=PUBLIC.ZZZ_BIKE_HOUR_WEATHER2 out=work.tds equals;
  by address Date_Time;
run;

/* Adding new column for total activity */
data WORK.TDS1;
  set WORK.TDS;
  by address Date_Time;
  /* Adding the change values to get total activity per hour in a station */
  Total_Activity = Avail_Bikes_Increase + Avail_Bikes_Decrease;
run;

/* Creating new column for accumulative activity in a station */
data WORK.TDS2;
  set WORK.TDS1;
  by address;
  /* For 1st appearance of address value set to 0 */
  if First.address then Accum_Activity = 0;
  /* Adding each quantity of activity and storing in the new column*/
  Accum_Activity + Total_Activity;
  /* Inserting new column as index of row per station */
  if First.address then Count = 1;
  /* Increase value by 1 for each row for a station */
  Count + 1;
run;

/* Save last value per station as the activity and frequency */
data WORK.TDS3;
  set WORK.TDS2;
  by address;
  /* Take value for the last appearance of a station address in the data */
  if Last.address then do;
    /* Save summed activity to new column*/
    Station_Activity = Accum_Activity;
    /* Save count of rows per station to new column */
    Station_Freq = Count;
  end;
run;

/* Re-sort data by address and then by last hour - to check summed values */
proc sort data=WORK.TDS3 out=WORK.TDS4 equals;
  by address descending Date_Time;
run;
```

```
/* Change nulls in total activity and frequency values to be correct values */
data WORK.TDS5;
  set WORK.TDS4;
  by address;
  /* Create temporary values to save summed values */
  retain sum_act freq;
  /* Assign temporary activity value per address */
  if First.address then sum_act = Station_Activity;
  /* Change all null values to equal activity totals */
  if missing(Station_Activity) then Station_Activity = sum_act;
  /* Assign temporary frequency value per address */
  if First.address then freq = Station_Freq;
  /* Change all null values to equal frequency totals */
  if missing(Station_Freq) then Station_Freq = freq;
  /* Remove temporary values */
  drop sum_act freq;
run;

/* Re-sort data by address and then by datetime */
proc sort data=WORK.TDS5 out=WORK.TDS6 equals;
  by address Date_Time;
run;

/* Remove unnecessary columns and add weight column */
data WORK.TDS7;
  /* Remove the unnecessary columns */
  set WORK.TDS6 (drop = Accum_Activity Count);
  by address;
  /* Take weight as rounded value of activity over frequency to 2 decimal places */
  Weight = round(Station_Activity/Station_Freq, 0.01);
run;

/* Re-sort data by datetime and then by station number */
proc sort data=WORK.TDS7 out=WORK.TDS8 equals;
  by Date_Time number;
run;

/* Using SQL to sum total bikes per each hour */
proc sql;
  /* Create duplicate table of data */
  create table WORK.TDS9 as
  /* Select all columns and sum available bikes per hour */
  select *, sum(Available_Bikes) as Total_System_Bikes
  from WORK.TDS8
  /* Grouping the data by the hours in each day */
  group by Date_Time;
quit;

/* Renaming and promoting hour grouped data with added variables to public server */
proc casutil;
  load data=WORK.TDS9 outcaslib="public" promote
  casout="ZZZ_Bike_Hour_Weights2";
run;
```