**[1] Probabilistic Forecasts of Bike-Sharing Systems for Journey Planning**[**https://dl.acm.org/doi/10.1145/2806416.2806569**](https://dl.acm.org/doi/10.1145/2806416.2806569)

1. It is not only importand, that there is a bike available at a station, but also that there is a free spot available when the user arrives at the destination station.
2. I don’t necceceraly want to know how many bikes are at a station at a given time, but the probability that a bike is available
3. Each station i = 1, … , N has a fixrd capacity ki, this is the maximum capacity, hence the maximum number of bikes that might be available at the station. The number of bikes currently available at station i at time t is noted as Xi(t) E {0, … , ki} and the number of availability for all station X at time t is X(t) E {X1(t) , …. , Xi(t)}. So each station has 2 states at the base level: if Xi(t) > 0, a bike for travel is available and Xi(t) < ki, a bike can be returned to that station. So, if a bike is taken the number of bikes decreases ( Xi(t) – 1 ) and if you bring a bike back to a station Xi(t) + 1.  
   What I will try to find out is the probability that a bike will be at a station at t + h, where h is the time added when the user needs the bike at a station, we also need to consider an available free spot at the destination station j so that all requirements Xi(t + h) > 0 and Xj((t+h) + traveltime) < kj are satisfied. This is, however, just the minimum requirements, the fluctuation of bikes also have to put into consideration so when the system gives the user a prediction, the number of bikes predicted to be available bikes or free spots should be higher than one. So the stochastic probability would be :

P(Ft | Xi(t+h) > 0 ^ Xj ( t + h + traveltime) < kj ))

1. Uses Markovian model, which means that I prediction is not based on the past data but the probability that a bike is available based on the current state of the stations. Those models have been succecfully tested in [7,10,12]. There is also the possibility of interconnecting the bike stations on a probabilistic basis in a way that the probability of a future time could be higher if bike stations nearby have currently more or less bikes, which indicates that in the close future some bikes will be brought back. This however would complicate the model immensely for very little gain as shown in

**[2]**

**[3]**

**[4]**

**[5]** **The-Bikeshare-Planning-Guide-ITDP:**

Dockless bikes are equipped with GPS so that the user does not need to find a specific station. This can however lead to the complication that in an area where there is not a high fluctuation of bikes you don’t find as many. With a fixed station you could potently have a more reliable source of available bikes then.

Several cities have made efforts to improve the ease and convenience of multi-modal trip

making by better integrating their bikeshare system with public transit. Operated by the transit

agency, Los Angeles Metro Bikeshare allows users to check out a bike using their Transit Access

Pass (TAP) card. Helsinki’s City Bikes system will be integrated into the mobility as a service

(MaaS) Whim app, which offers streamlined access to taxis, public transport, shared vehicles,  
and, soon, bikeshare through pay-as-you-go or monthly plans.1

**[6]**

**[7]**

**[8]**

**[9]**

**[10]**

**[11]**

**[12]**

**[13]**

**[14]**

**[15]**

**[16]**

**[17]**

**[18]**

**[19]**

**[20]**

**From my Programming**

I started off with some sample data which seems like it was a snapshot from all stations at a given time : X(t). First I just printed out the data complete as a Json to get a genrell understanding of the structure of the data which was basically one big column of countries which included a list of nested dictionarys. My first thought was to get a good dataframe which I can use for my purpose. My first attempt was to set the indexes as countrynames and then all parameters as column names, since the citys and also the places inside those countries were also just nested dictionarys I changed the approach and extracted just the stations, since this is what I am aiming for anyway.