Data Analytics - Project Stage 1

Jemma Herbert - 430147760

August 23, 2017

1 Business Understanding

1.1 Determine business objectives

1.1.1 Background

PaddlersPredictions is a service which predicts river levels, somewhat like a weather forecast. It makes money by selling subscriptions to this service. The target market is white water kayakers living in NSW, Australia.

Unlike in many other countries, in Australia white water kayaking is a logistically difficult hobby. The river levels are rarely high enough to be paddled, and when they are high they rarely stay high for very long. In order to plan their weekend adventures, kayakers require a least a day or two of advanced knowledge that the river levels will be sufficiently high. PaddlersPredictions provides this advanced warning about when the river levels will be high enough to paddle.

1.1.2 Business Objectives

It is predicted that PaddlersPredictions will sell more subscriptions if the predictions are more accurate. Most importantly it needs to predict whether a river is 'paddleable', ie. above some minimum threshold. Further precision about the exact level is much less valuable. In order to be useful, these predictions need to available and accurate several days in advance.

1.2 Business Success Criteria

This exercise will be judged as successful if it can generate accurate river level predictions at least 20% of the gauges in NSW, or at least 1 gauge that corresponds to a good white water section. The model should predict whether or not the gauge will be above the given threshold with at least 50% greater accuracy than a raw guess (ie. always no, always yes, or 50/50, whichever is closest to its history). The forecast should meet these requirement at least 2 days in advance (ie. a Saturday forecast provided on Thursday morning).

These targets are strict but necessary. Such high accuracy is necessary because it is very inconvenient to a paddler to make plans to go kayaking, but the river is not

high enough on the day. It is necessary to be predicting at least 2 days in advance because the kayakers need enough time to gather a party and get organised. Ideally this project would create an accurate model for every gauge in NSW, but this is not strictly necessary. If this project is able to predict even just one gauge corresponding to a white water section it will be a success.

2 Asses Situation

2.1 Inventory of Resources

2.1.1 Software and Hardware Resources

Several software options are available, including:

- Matlab
- R
- Python
- Excel
- SQL

The hardware available is limited to a single, middle of the range, laptop. External storage is available but the transfer speeds are limited to USB3.

2.1.2 Data and Knowledge Stores

Vast amounts of publicly available data are relevant to this project. These data sources include: the Bureau of Metorology (BoM), the NSW Office of Water (OoW) and several kayaking specific websites. The links to these websites are `http://www.bom.gov.au/climate/data/index.shtml`, `http://realtimedata.water.nsw.gov.au/water.stm` and `http://www.kayakcanberra.com/heights/` respectively.

The data on the BoM and OoW can be downloaded on a 'per-site' basis (as in, a separate link to a single file for each gauge or weather station across the state) for free, or can downloaded in bulk for a processing fee. We do not plan to purchase this data because the $95 fee exceeds the strict budget of $0. Instead a web-crawler will be written to download each file individually from both websites.

The data available at kayakcanberra.com gives the minimum kayakable level at 25 gauges across NSW. This data is available as text on their website. It includes the name of the gauge, the minimum level, and a link to a short history of the levels at the gauge.

The BoM and OoW have huge quantities of data available. At each collection point across the state (918 river gauges and 5030 rainfall monitors) the full history of data collected is available. At some sites this history extends back, every day, for almost 150 years. This is in csv format, inside a zipped folder. It contains, date and time that the data was collected, the gps coordinates of the collection point and sometimes some sort of judgement as to the quality of each data point.

2.1.3 Personnel resources

This project needs to be completed within 2 months by a single student, this comes to about 80 man-hours of labour available. Furthermore this student is only just starting to learn about data analytics. This should be achievable.

2.2 Requirements, Assumptions and Constraints

2.2.1 Requirements

It is required that the stages of the CRISP-DM cycle be completed by certain dates. Roughly, each stage must be completed in about 3 weeks. It is also required that the BoM and OoW allow my crawler program to collect all of their data automatically, this is a task that is too large to be done by hand.

2.2.2 Assumptions

It is assumed that the character of the rivers has remained relatively constant over the last century and a half. This is not always true, eg some dams have been built, but data is not available about these changes. It is also assumed that the collected data is accurate. Most importantly the timestamps, river levels and rainfall are relatively accurate. Other factors, such as GPS coordinates can afford to be less accurate.

2.2.3 Constraints

This project is constrained by lack of accessible data on geography of catchment areas. It is also constrained by the lack of knowledge and experience of the personnel involved.

2.3 Risks and Contingencies

Risk: The data analysis will take longer than the available time.
Contingency: more time can be devoted to the project by taking time away from the student's sleep.

Risk: The data may be of poor quality which would make it difficult to clean and analyse.
Contingency: The problematic data could be removed and the analysis performed on a smaller subset of the data. Since there is so much data available it is not a problem to discard a large portion of it if necessary.

Risk: There is no discernible pattern from the data and the river levels cannot be accurately predicted.
Contingency: The tool can be made to always predict that ever river will be too low to paddle and this would be correct about 90% of the time.

2.4 Terminology

Level: River height at that gauge. This is not an indicator of flow rate but merely depth at the point in the river. The same river will have different levels at different gauges.

Station: A location where rainfall data is collected. In some cases a gauge and station will be the same physical place, but they are treated as separate entities in this analysis.

Paddleable: The section of river near the specified gauge is above some specified minimum threshold. This threshold is determined by concensus of the kayaking community and is reported on several websites such as kayakcanberra.com.

2.5 Costs and Benefits

The costs for this project are very small. The data is all available free from government sources off the internet. The analysis is being run off existing hardware with free, or student-versions of software. The only significant cost is labour of the student and this is very, very cheap.

The benefits of this project are potentially significant. If the final product can meet the goals of giving a 90% accurate level forecast 3 days in advance then a cheap online subscription would be purchased by every white water kayaker in NSW. Unfortunately white water kayakers are typically stingy and so the subscription would need to be exceptionally cheap. If all 500 members of the Canberra White Water Facebook group (estimated 50% of total population of NSW white water kayakers) were to purchase the subscription at \$5/year, it would profit \$2500/year. Compared to the very small initial cost of time this is a worthy investment.

3 Data Mining Goals and Success Criteria

This is a *prediction* problem. The goal is to predict the future river heights, at the gauges, given the forecasted rainfall. The prediction should be at least 50% more accurate than a guess, and available at least 2 fulls days in advance. The goal is to have accurate predictions for every river gauge across NSW. The project will still be considered successful if it can meet these goals for at least 20% of the NSW, or at least 1 (of the 25) good white water section indicators. These good indicators are listed on kayakcanberra.com.

For example, on Thursday morning it can predict whether the gauge level for the Murrimbidgee near McDonald is at a paddleable level on Saturday. The accuracy requirements of this guage are dependent on how frequently it is paddleable. If it is paddleable only 20% of the time, then it is required that the prediction is wrong no more than 10% of the time. This is a 50% improvement on a guess of always 'not paddleable'.

This project does not require that the model is deployable, or accessible to the general public. Deployment will be a future project once the model is developed and validated.

4 Produce Project Plan

4.1 Project Plan

Table 1:

Phase	Time	Resources	Risks
Business Understanding	3 week	Personnel	economic change
Data Understanding	2 weeks	Personnel, software, hardware, data	data problems, technology problems
Data Preparation	2 weeks	Personnel, software, hardware, data	data problems, technology problems
Modelling	3 weeks	Personnel, software, hardware, data	unable to find sufficiently good model, data problems, technology problems
Evaluation	1 week	Personnel, software, hardware	change in the requirements
Deployment	3 weeks	Personnel, software, hardware	inability to implement model effectively

4.2 Initial Assessment of tools and techniques

The tools and techniques used will depend on the skills of the analysts. The analysts currently know how to use software such as Excel, Matlab and Python, however it may be necessary to learn additional software such as SPSS or R.

5 Data Understanding

5.1 Initial Data Collection Report

The data was collected from 3 primary sources; the Bom, the OoW and kayakcanberra.com. The data from kayakcanberra was collected by hand, as it was only 26 lines. This data was copied into an excel file. The data from Bom and OoW was much larger and more difficult to collect. Two web crawlers were written to systematically browse the BoM and OoW website and download the relevant files. These crawlers were written in Python and the source code for one is included in Appendix A(the other crawler is very similar). For each website a list of all sites was available and a zip containing all the data for each site could be accessed.

Due to the data collection method, many thousands of files will need to be merged into a single data source. This should not be problematic as all the individual files are

highly consistent.

The large quantity of data available will hopefully mean that accurate predictions can be made. If accurate predictions cannot be made it is likely that this would be due to lack of breadth in the data, rather than a lack of volume.

Almost every file contains large chunks of missing rainfall or gauge level values. This occurs because the dates are listed from the start of the calendar year when data was first collected. These values will need to be removed when the data sources are merged. No other missing values have been found in the data, but an extensive analysis has not been conducted so this is not certain. If other missing values occur they can also be omitted, since there is plenty of complete data available.

5.2 Data Description Report

The format of the BoM and OoW data is thousands of separate zip folders containing a .csv file. For the BoM data, each folder also contains a .txt file describing some features of the collection station and how to interpret the data. The kayakcanberra.com data is a single .xlsx file.

The kayakcanberra.com data has the following properties:

- Number of rows: 26 million
- Number of columns: 4
- Data columns:
  - River gauge name - string
  - Current river level at gauge - measured at time of collection, in meters, given 2dp
  - Minimum - paddleable threshold at that gauge, given in meters, given 1 or 2 dp
  - Time of reading - data/time of last updated gauge levels

It is expected that the useful columns from this source will be the gauge name (called 'river name' in the data) and minimum paddleable level (called 'minimum' in the data). The other columns including data about the river level at the time when the data was collected is largely irrelevant.

The BoM data has the following properties:

- Total number of rows: 83 million
- Number of stations: 5030
- Number of columns: 8

- Other data: meta-data
- Data columns:
  - Product Code - 10digit alphanumeric
  - BoM Station Number - 6digit number
  - Year
  - Month
  - Day
  - Rainfall amount - positive number, measured in millimetres
  - Period over which rainfall was measured - positive number, can be fractional, measured in days
  - Quality - Y/N indicating whether the data has completed a quality control process or not
- Meta data:
  - Station name - a string of characters, unique to that station
  - Year site opened - a date of varying formats, sometimes only year or month/year
  - Year site closed - a date of varying formats, sometimes only year or month/year. Sometimes not included.
  - Latitude - number, measured in decimal degrees, south, negative, given to 2dp
  - Longitude - number, measured in decimal degrees, east, positive, given to 2dp
  - Height of station above mean sea level - number, measured in metres
  - State - 3 letter character code

It is expected that the most useful columns will be rainfall amount, and date. Latitude, longitude, elevation and quality might also be somewhat useful.

The OoW data has the following properties:

- Total number of rows: 7.9 million
- Number of stations: 644
- Number of columns: 5
- Other data: meta-data
- Data columns:
  - Date - day/month/year

  - Total Rainfall - measured in millimetres at the gauge
  - Quality of rainfall data - integer (1-255), code corresponding to description of data quality in metadata
  - River Level - positive number, measured in meters, 3dp
  - Quality of river level data - integer (1-255), code corresponding to description of data quality in metadata

- Meta data:
  - Site Number - 6 digit numeric code,
  - Site Name - character string
  - Latitude - number, measured in degrees, south, given to 4dp
  - Longitude - number, measured in degrees, east, given to 4dp
  - Elevation - number, measured in meters, given to 3dp

It is expected that the most useful columns will be river level, and date. Latitude, longitude, elevation and quality might also be somewhat useful.

6 Bibiliography

IBM, 2011. *IBM SPSS Modeler CRISP-DM Guide* `ftp://ftp.software.ibm.com/software/analytics/spss/documentation/modeler/14.2/en/CRISP_DM.pdf` (Accessed 23/8/17).

A Python Web Crawler Code - BoM

```
print("Importing libraries.")
from selenium import webdriver
from selenium.webdriver.support.ui import Select
import time
import csv
from os import listdir
from os.path import isfile, join
import numpy

def print_log(message,priority):
        threshold=2
        if priority>=threshold:
                print(message)
def make_file(river_num):
        base_path="C:\\Users\\Jemma\\Documents\\Uni\\sem2_2017\\INFO3406\\Assignment\\
            ↪ rainfall_data_collection\\Data"
        file_name='failed_'+river_num+'.txt'
        open(base_path+'\\'+file_name, 'a').close()
        print_log('failure file created',2)

def download_station(station_list):
        print_log("Starting the browser.",0)
        driver = webdriver.Chrome()
        done_count = 0
        for station_num in station_list:
                try:
                        driver.get("http://www.bom.gov.au/jsp/ncc/cdio/weatherData/av?
                            ↪ p_nccObsCode=136&p_display_type=dailyDataFile&p_startYear=&p_c
                            ↪ =&p_stn_num=" + str(station_num))
                        print_log("Loading the page.",0)

                        myLink=driver.find_element_by_xpath("//a[@title='Data file for daily
                            ↪ rainfall data for all years']")
                        myLink.click()

                        time.sleep(1)
                        done_count+=1
                        print("done " + str(done_count) + " out of " + str(len(station_list)))

                except Exception as e:
                        print_log("An exception occured:",1)
                        print_log(e,1)
                        print_log("Station number \%s has not been sucessfully downloaded." \%
                            ↪ station_num,2)
                        make_file(station_num)
        driver.close()

with open('rainfall_station_data.csv', 'rU') as f:
        reader = csv.reader(f)
        your_list = list(reader)
all_stations = [item[0] for item in your_list]

#Check against downloaded files
mypath="C:\\Users\\Jemma\\Documents\\Uni\\sem2_2017\\INFO3406\\Assignment\\
    ↪ rainfall_data_collection\\Data"
downloaded_stations = [f.split('_')[1] for f in listdir(mypath) if isfile(join(mypath, f))]
not_downloaded=[station for station in all_stations if station not in downloaded_stations]

print(str(len(downloaded_stations))+' stations have been downloaded. \%d stations remaining.'
    ↪ \%len(not_downloaded))

# split into chunks
num_threads=2
chunks = [list(i) for i in numpy.array_split(numpy.array(not_downloaded),num_threads)]

from multiprocessing.dummy import Pool as ThreadPool
pool = ThreadPool(num_threads)
pool.map(download_station,chunks)
```

Project Stage 2 – Summarise and Analyse Data

Data Understanding

Data Exploration Report

Hypotheses

Hypothesis 1: If the river is not significantly above its average level, then recent rainfall will result in increasing river levels. The amount that the river level increases will be monotonically increasing function of amount of rainfall, ie. the more rain, the higher the river. (Recent = within 10 days. Specific locations = in that river's catchment, usually to the west of the gauge since most rivers in NSW run west to east.)

Hypothesis 2: As the river levels rise, more rainfall is required to raise the level by the same amount. If there is a comparatively small amount of rain (compared to that which originally caused the river to rise) the river level will fall toward its average level.

Hypothesis 3: Rainfall within the catchment of a gauge will directly affect the gauge level. Rainfall outside its catchment will have a less strong correlation. Catchments are generally to the west of gauges (since most rivers in NSW run west to east), but can also be large distance north/south.

Hypothesis 4: River levels will respond more quickly to rainfall that is a short distance away from the gauge and more slowly to rainfall (within its catchment) farther away from the gauge.

Hypothesis 5: Some gauges are on the same river, so gauge A will have a similar profile as gauge B but slightly delayed since one of them must be downstream from the other. Significant tributaries or weather events in between the gauges will result in different profiles.

Hypothesis 6: Gauges will generally be higher in the wetter months of the year (September – December)

Promising Attributes

It is expected that the attributes which will best predict the river level at a specific gauge will include:

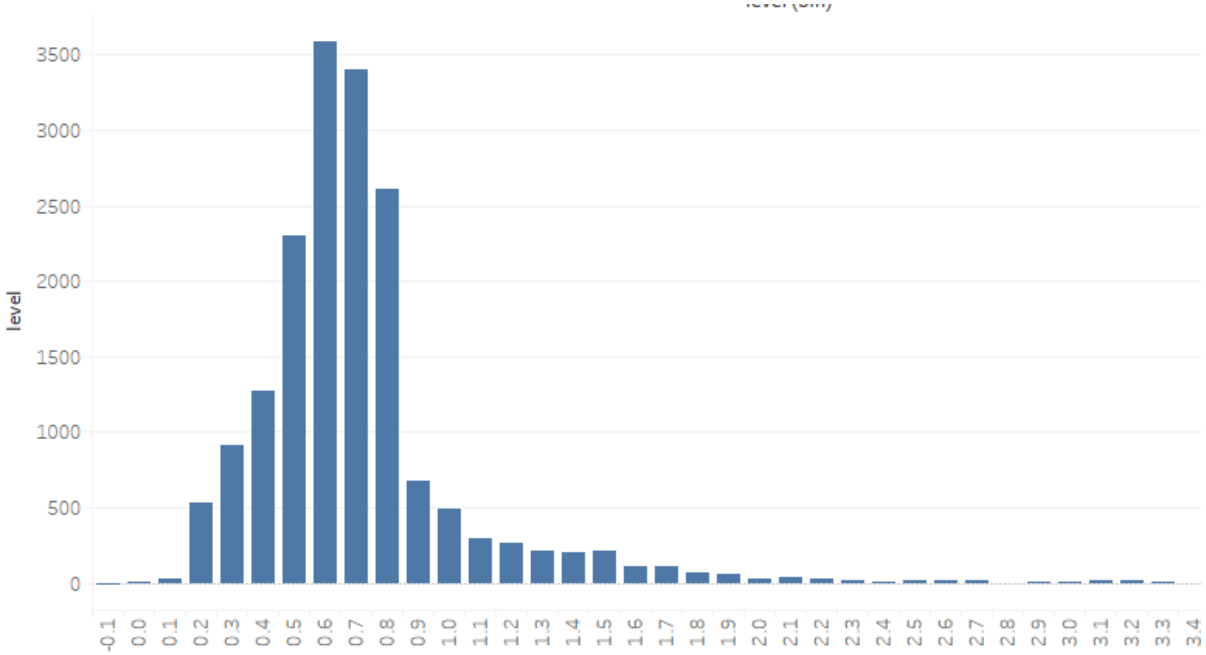
- The level at that gauge in the past few days (specifically how much above its 'baseline' level it is)
- Rainfall in the last 10 days at some specific gauges
- The current month of the year

Characteristics of the Data

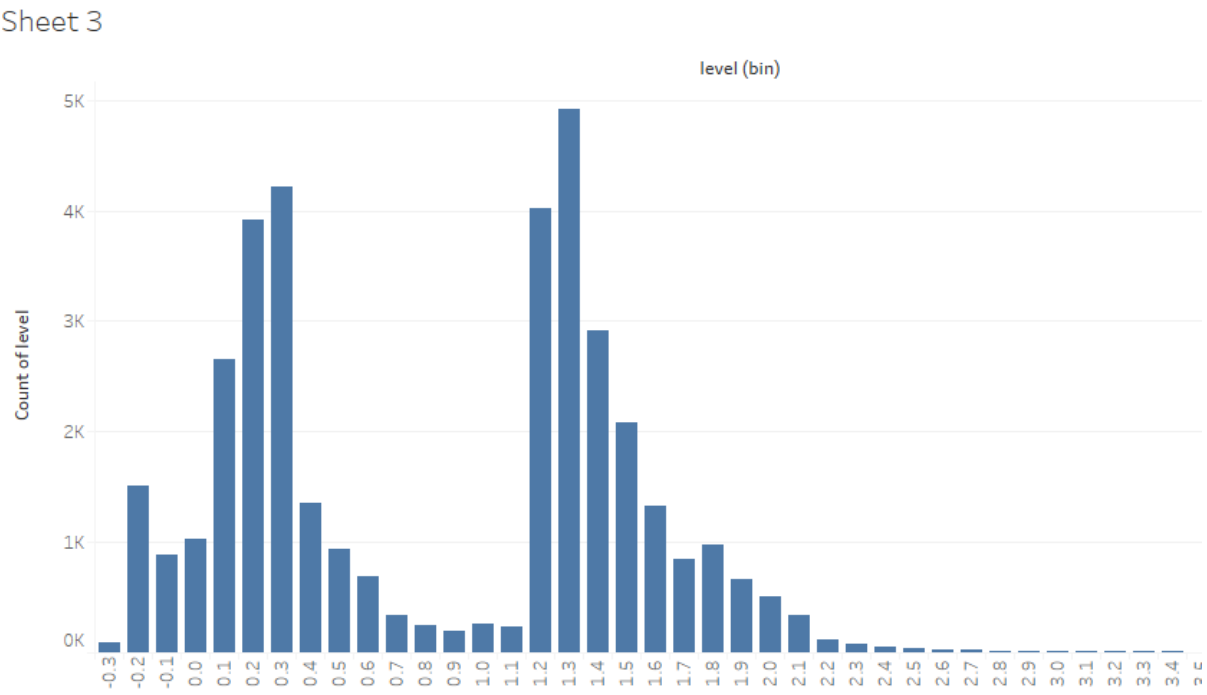
River level

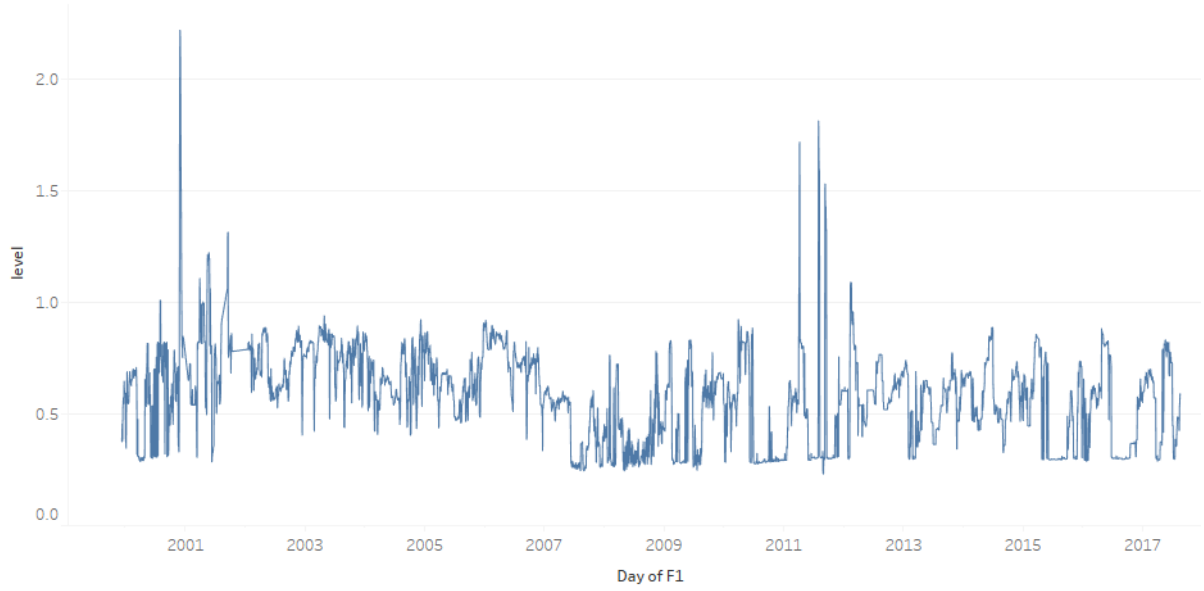
Due to the large quantity of data I was unable to look at the distribution of every single guage, however I made histograms and looked at the profiles for a random subset of the gauges. A typical gauge looks like this the following figure. There is a small range of levels which are very common

(the baseline level), and there is a small left tail where the river occasionally drops slightly below its baseline and a very long right tail where the river occasionally rises far above its baseline level.



One of the gauges that I viewed had a profile with a double peak. It was found that this occurs because the river is in the snowy mountains and it has steady but different summer and winter levels.





This plot shows the change in river level at a single gauge over time. It clearly has peaks and a 'baseline' level which it falls back to.

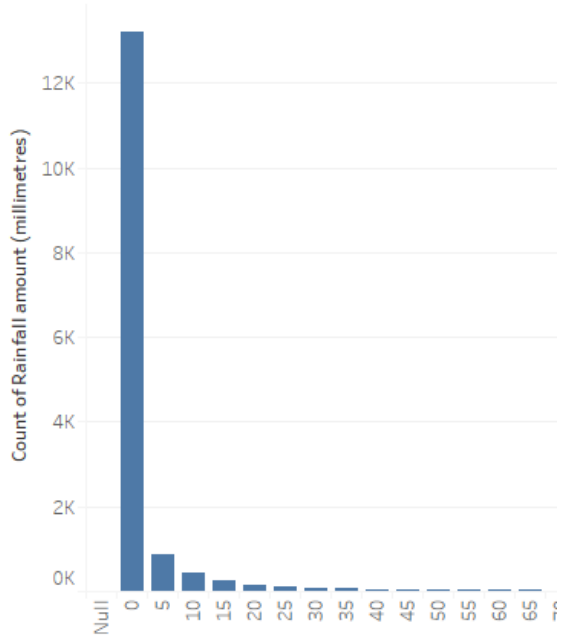
It was expected that no river level would ever have a negative level, however this assumption was wrong. 243 rivers read negative levels, and some more through research shows that these are valid values. Some gauges measure 'zero' from an arbitrary point on the shore, and the river level can fall below this point.

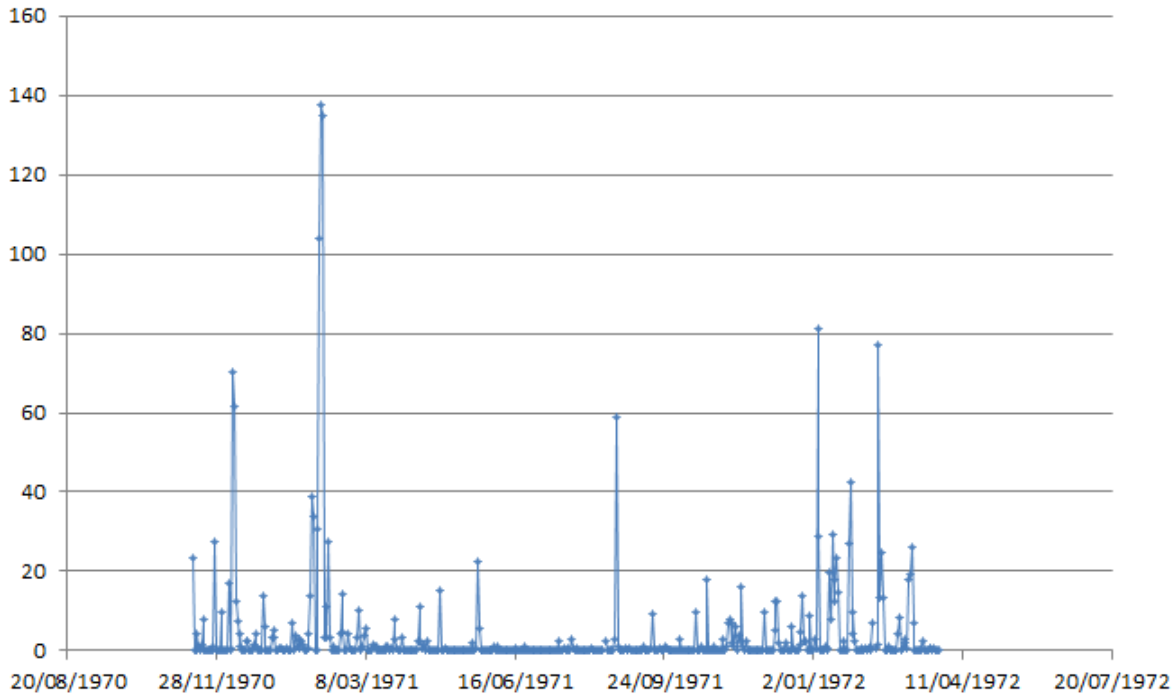
Rainfall

Similarly to the river levels, this was too much data to plot, or to look at manually. Instead a small random sample was assumed to be representative of the whole.

This is a histogram of amounts of rain at a typical station. It has a very long tail due to freak weather event where huge amounts of rain fall in a single day. It also has a large spike at 0 because most days it doesn't rain.

A typical rainfall over a couple of years of shown in the following figure. There are periods of many large spikes (rainy days), and then there are periods with very few small spikes, this shows the significance of the seasons.





Meta data

All Gauges have a station within 4km. So should be able to get good estimate of rainfall at every single gauge. However not all stations are nearby gauges.

Data Quality Report

Missing Data

There is lots of missing data in this data set. Some of it is meaningful some of it is just a quirk of the system.

Rainfall: 10 964 911 rows (13% of the set) of the Rainfall table contain a date but no rainfall data. These points are useless and should be removed. These values occur because every file begins with a consecutive set of blank values. Regardless of when in the year data started to be collected, the dates are listed from the first of January. So if data collection started in November there might be 300 null values corresponding to every day in the first 10 months of that year.

Rainfall Period: This this meant to show the period over which the listed rainfall was collected. This data is more useful when the extracted data is aggregated by month. The blanks in this column correspond almost always correspond to zero rainfall (only 115 instances corresponding to non-zero rainfall). Almost every period is '1' except for a few hundred exceptions which are 2 or 3 days, and 14 observations that are longer periods than that (up to 17 days). There are so few of these non-one periods that it can be assumed that every value (including the nulls) are all 1.

River Levels: 823 676 rows (10% of the set) of the Levels table contain a date but no level data. These points are useless and should be removed. These points occur where rainfall at the gauge (not used for this dataset) is measured, but gauge level isn't. Some of the more extreme cases have 5 points of level data in 1947, and then no level data for 60 years, then 15 more points of level data in 2007 before the gauge closes.

Meta data: The stations meta data was perfectly complete, no value was missing. The meta data from 29 gauges is totally absent. This meta data could be found by going back to the original source but it is too much work, so instead the data from these gauges will be discarded. The data from these 29 gauges corresponds to 240958 out of 7054076 (3.5%) total river level observation.

Data/Measurement errors:

Several plausibility check were done to ensure that the data is correct.

No gauge should every read more than ~2m. There are 2 rivers that average significantly higher than this, on the Hawkesbury at 6m and 15m. In addition there are occasional readings on other gauges that are significantly above this level, but that is acceptable since floods do occasionally occur.

Rainfall should never be negative: This is true, there are no negative rainfalls listed in the data set.

Rainfall amount should never be more than 1090mm in a day: This is the record for most rainfall in a day in Australia. This fact holds true in the data set.

Coding Inconsistencies

River levels data is encoded in a separate csv file for each gauge. Amongst gauges there are two different formats of the csv file. One format lists rainfalls at the gauge (not used, rainfall for this project is obtained from the BOM) and has 5 columns (date, rainfall, rainfall quality, level, level quality). The other format doesn't list rainfall but list maximum and minimum river levels in that day. It has 7 columns (date, mean level, mean quality, min level, min quality, max level, max quality). This means that when reading in the files, the format needs to be detected and accounted for.

The value 'river level qualities' is a set of codes that correspond to different ways in which the data was measured and checked. These codes are complex and non-linear, a higher value doesn't always correspond to lower quality data. Some of these quality ratings do not give specific information

about the quality of the readings just extra information like 'was collected pre-1973'. The useful component of this data are the codes '255 = data unavailable' and '201 = no data exists'

The rainfall quality measure is mostly consistent ('y'/'n') indicating whether or not the data has been checked before logging. 97% of records are 'Y', they have been checked. 3% of records have not been check and should be removed to avoid misleading data. 888 (<0.1%) of records are blank. This is such a small portion of the total data that these records can be assumed that they weren't checked and discarded.

Bad Metadata

The metadata was of the gps coordinates of each station and gauge. This data was plotted onto a map of NSW to ensure its accuracy. This data seems to be very accurate. A close look will show that all gauges lie on rivers and all stations lie on land. Nothing stations or gauges are at sea or outside the border of NSW.

Data preparation

Select Data

This is a very large data set so we can afford to choose only the cleanest and more effective data.

Excluding Attributes

State: the state is the same in every single record. This will not help discriminate differences in the data set so should be removed.

Level Quality: The river level quality codes are too complex and inconsistent to be useful. Note: this attribute should only be discarded after it is used to discard poor quality items.

Rainfall Quality: The rainfall quality is too imprecise to be very useful. All poor quality observations will be discarded and all high quality observations will be kept. Once all poor quality observations are discarded every observation has the same value 'Y' so this attribute is no longer useful.

Open/closed date: The opened and closed date of stations is not important. It is only important to know for which dates data is available. This is kept it the 'date' field of every observation.

Period of rainfall: Only a very small portion of the total rainfall observations have a period of anything other than 1. 246000 observation, 0.35% of the total rainfall data set. In addition to the observations which have periods>1, there are 115 observations which have no listed periods. Since these are such small portions of the data, almost everything is 1 so this variable is not useful.

Excluding Items

In addition to the attributes which will be discarded, there are also several sets of rows that were discarded. All rows which satisfied any of the following conditions should be discarded.

- **Null rainfall** - (10964911 nulls; 72170824 not nulls) ~ 13%. It is not useful to know the dates when a rainfall observation was not made.
- **Null level** - (823676 nulls; 7054076 not nulls) ~10%. It is not useful to know the dates when a level measurement was not made.

- **Observations taken at stations > 100km from the nearest gauge** (Lord Howe and Norfolk Islands). These stations are so far away that their weather could be very different to the weather near the stations, therefore it is not useful.
- **Rainfall where quality = 'N' or ' '** – this only constitutes 3% of data and it could cause problems if they are drastically wrong. This 3% of the data is distributed across 33% of stations. This means that removing it will not result in any big blank sections where no rainfall data is available.
- **River levels at gauges which don't have meta-data about location**. - Location of gauges is very important to correlating it with nearby stations and gauges. It is no use to know that some gauge somewhere in the state is high because that information is too generic to tell us anything about a specific gauge that is being predicted.

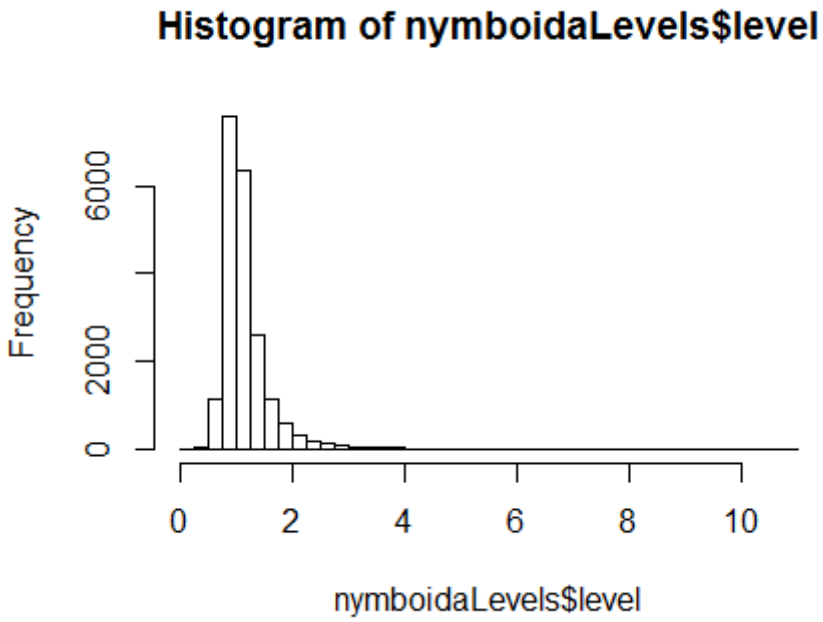
Some data was considered for discard but decided against it.

- Could discard all data from stations > 50km from the nearest gauge because it doesn't directly influence any river levels. Didn't do this because all these points lay in the north west of NSW and weather tends to move west to east, so this weather could be a predictors of future weather further east, and hence future river levels.
- All river level data with quality codes of 255 or 201 (very poor or no data) were going to be discarded. None of these records remained in the database after the previous stages of cleaning.

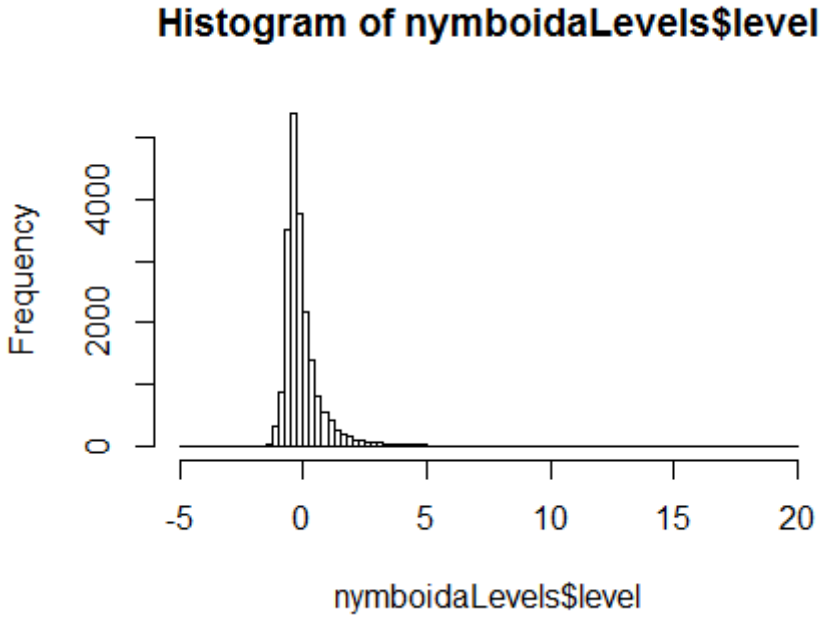
Clean data

The river level data should be normalised so that it is easy to compare between gauges. The zero of a gauge is arbitrary and the level is not a good indicator of flow rate. Two different gauges reading the same state of a river may have very different values. Similarly they may vary by different amount to the same increase in flow rate dependent on the profile of the river at that exact point on the river. This applied to gauges where zeros are not meaningful. It does not apply to rainfall where zeros are meaningful.

Calculate and store the mean and standard deviation for each gauge. They are stored so that the actual value can be retrieved later. The normalised value is given by new = (old-mean)/sd. The distribution of river levels is not perfectly normal, it has a long right hand tail, since there is no limit on how high the water can get, but it cannot get below minimum.



Histogram of Nymboida river level data before being normalised



Histogram of Nymboida river level data after being normalised

Construct Data

Derive attributes

A possibly useful attribute to derive would be the distance from each gauge to each other gauge. This would be calculated as sqrt((lat1-lat2)^2 + (long1-long2)^2))*111.1km. The trouble with adding

this attribute is that it would blow up the dimensionality of the final single data table. Without the attribute the final data table might have 8 or 10 dimension, but if every observation were to also list the distance to every other gauge it would need another 615 dimension (one for each gauge). This is too messy and would make dealing with the vast amounts of data even harder than it already is.

Generate records

The gauge level could be predicted to be half way between adjacent points if a single data point is missing. This is because most rivers change level relatively slowly (a period of several days) so it is unlikely that the estimated point is significantly different to its neighbours. This would not work if the river is fluctuating very rapidly, or if there are several consecutive data points missing.

Integrate

The sourced river levels data was a collection of csv files. Each file corresponds to a single gauge and contains some meta-data about that gauge. Initially all the meta-data was extracted from the files and inserted into an sqlite3 table. Then the levels data itself was extracted and inserted into a separate sqlite3 table.

Similarly with the rainfall files, the meta-data extracted from the associated text files was inserted into a separate table to the actual rainfall data. SQL was used because it is easy to formulate specific queries like “how many records have no rainfall recorded?” or “how many stations are >200km away from the nearest gauge?”

The plan was to clean and analyse the data using SQL and then export that data to R for further analysis. The cleaning was successful, easily performing operations like deleting any observations that occur in stations that are located >200km from the nearest gauge. However I failed to export this dataset to R. Several approaches were tried and failed for different reasons.

One approach to getting the database into R was to get R to directly read the .db file. This approach failed because I could not load the RSQLite package. Another approach I tried was to export the database into a csv file using python then load that csv file into R. This failed because my computer ran out of memory to store such a large variable (the list of all observations) in memory. I finally tried to import all the raw csv files from the original source folder, unmodified by sql. This failed because it took too long, the process didn’t ever complete.

Being unable to import all my data into R to analyse it, I decided to import a subset of representative data to demonstrate the theory and hopefully in the coming weeks I can get some help to import the rest of it. Most of the operations used in the R analysis could be used on a dataset 1000x larger without problems.

The dataset I imported included: 1 gauge at the Nymboida river, 6 nearby rainfall stations, all the gauge meta data and all the station meta data. This data imported into R was the raw data (unmodified from the source) so it had to be re-cleaned in R.

The gauge meta-data and the river level data were joined based of gauge-id. Similarly the station meta-data and the rainfall data were joined on station-id. It is also intended to join rainfall and river levels by adding an extra variable such as ‘type’ which indicates whether that data is a rainfall-station data point or a river-level-gauge data point.

Formatting Data

It is expected that some sort of machine learning algorithm will be used to produce the model. This is because the data set is so large so it should work, and only the model only needs to produce predictions so understanding of the model is not necessary.

Machine learning algorithms require numerical or categorical data. They probably can't easily deal with dates, so the dates will be converted into 3 separate columns (year, month, day). This has the added benefit of making 'month' its own variable. It is expected that month may have a significant impact on the expected river levels because of the seasons.

As discussed in previous sections the river levels data was normalised about zero. The normalised and raw histograms are shown on a previous page.

The data should all be compressed into a single data table containing all of the data from the 4 natural tables (rainfall, levels, gauges meta data, stations meta data). This is necessary because there are very few modelling techniques that can meaningfully take data from multiple source.

Dataset Description

As discussed above, I cleaned the full dataset in sql/python but then failed to import that data in R. Only a small subset of the full dataset is currently in R, but I am hopeful that in the future the full dataset that is currently in SQL will be imported into R also. As such both datasets are described here.

Dataset in R:

All of the following data is contained in a two R data frames, one for rainfall and one for levels. The levels data contains 20534 rows and 7 columns. This corresponds to only a single gauge from the original set, due to technical issues. Hence all the gauge information is duplicated in every row of the data frame. There are no nulls in any field.

Levels data frame:

- Date of observation – integer (eg. 00:00:00 16/06/1956). All times are at 00:00:00.
- Level – double, normalised about zero. A level of 1 indicates it is 1 standard deviation above the mean.
- Gauge id – double, the id of the gauge at which the measurement was taken
- Name – integer, the BOM name of the gauge
- Latitude – double, the latitude of the gauge in degree east
- Longitude – double, the longitude of the gauge in degrees south
- Elevation – double, the elevation of the gauge above sea level in meters

The rainfall data includes information from the 6 gauges nearest to the chosen gauge. This is a very small subset of the total data in the SQL database. This table contains 11 528 008 rows and 8 columns.

Rainfall data frame

- Station id – integer, the BOM id of the rainfall station where the measurement was taken
- Year – integer, the year in which the measurement was taken

- Month – integer, the month in which the measurement was taken
- Day – integer, the day in which the measurement was taken
- Rainfall – double, the amount of rainfall measured, in mm
- Name – character, the BOM name of the rainfall station
- Lat – double, the latitude the rainfall station, in degrees east
- Long - double, the longitude the rainfall station, in degrees south.
- Elev - double, the elevation of the rainfall station, in meters

It is intended to combine these two data frames, but with this current subset of data it is not very meaningful to do so.

Dataset in SQLite/ Python

The main difference between this data set and the R data set is the volume of data. Since this data is stored in SQL it is not necessary to merge the tables because they can so simply be merged with a single command. Instead there are 4 separate tables.

The descriptions of each attribute are the same as above, so are not duplicated here.

Rainfall table:

- Station id - integer
- Year - integer
- Month – integer
- Day - integer
- Rainfall - number
-

River Levels table:

- Gauge id - integer
- year – integer
- month – integer
- day – integer
- level – number, not normalised

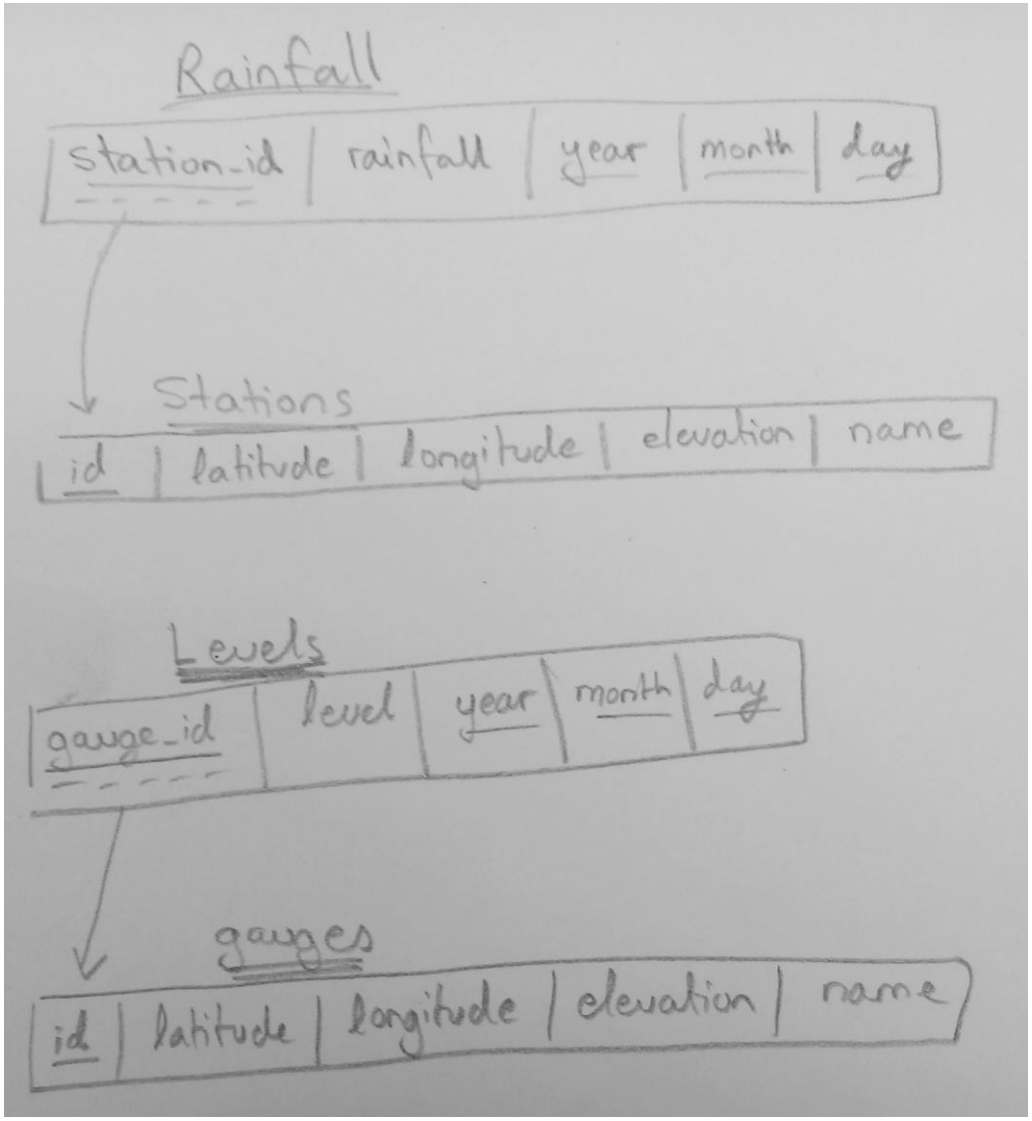
Gagues meta data:

- Gauge_id – integer
- Latitude – number
- Longitude – number
- Elevation – number
- Name – text

Stations meta data:

- Station id – integer
- Latitude – number
- Longitude – number

- Elevation – number
- Name - text



Report - Stage 3

Jemma Herbert (430147760) and Ke Henry Xu (450554065) - group #27

Modelling

Selecting Modelling Techniques

Training and Validation Sets

The data should be separated into training and validation sets. The sets should not be chosen randomly, due to the time-series nature of the problem. Instead the validation set should be a continuous chunk of time, preferably the most recent data. This is because we are not trying to predict the river as it was 30 years ago, for example, we are trying to predict the river now. As such we want to know how well the model predicts the levels now.

Quantity of data

Although there is a huge total amount of available data, this problem of predicting the river level for every river in NSW is hugely data intensive. If every gauge is to be trained independently then there needs to be sufficient amount of data available for each and every gauge. This is not the case, some rivers have very little data available (and some have very much data).

If we are to predict the levels of rivers which do not have much data then we will need to find some other data in the rest of the data set that helps predict those rivers. This could be achieved by clustering rivers that are correlated. Rivers would be correlated when there are multiple gauges on the same body of water. If we cannot find other data to predict these river then they will need to be discarded.

An error was made in the cleaning of the data which resulted in ‘losing’ the data from almost half of the original gauges. The original data set had data from 615 rivers, the final dataset had only 377 rivers. This error occurred when the levels data was combined with the rainfall data, joining on the nearest station to each gauge and also the date. Data was lost because the nearest station did not always have rainfall data for the same dates as the gauge had level data. Unfortunately this mistake was not realised until it was too late to remedy, so this analysis was conducted on this reduced data set.

This is the offending query that resulted in loss of data.

```
SELECT Level, Rainfall, L.year, L.month, L.day, gauge_id
FROM Levels as L JOIN Gauges ON (gauge_id = id)
JOIN Rainfall as R
ON (nearestStat=station_id and L.year=R.year and L.month=R.month and L.day=R.day)
```

The reduced data set has 2.9 million rows of data from 377 gauges. Each row contains: gauge_id, level, rainfall at the nearest station, date.

Data Quality

In order to accurately predict river levels we need accurate river levels data. Luckily, the data we have collected has quality ratings (indicating the accuracy of each data point), and low quality data was already removed in the data preparation phase. The data quality rating did not specify numerically how accurate the data was guaranteed to be, however it is plenty good enough for the kayakers who use it, so it should be good enough to make prediction for the kayakers.

Data Type

The available data types are:

- Numeric (river levels, rainfall)
- Catagorical (gauge_id)
- Date (day, month, year)

The river levels and rainfall data could also be considered as time-series, since they correspond with a sequence of dates. This will probably be the best way to interpret the data since it is expected that there will be a strong dependency is the last few days of levels and rainfall.

Interpreting the river levels and rainfall data over time as a time-series in R will be one of the biggest challenges. There are several approaches that could achieve this.

One approach to getting time series data is to use the SQL databases to retrieve 10 extra columns. 5 columns corresponding to the river level in the previous 5 days, and 5 columns corresponding to the rainfall in the previous 5 days.This was attempted using sql in R with a single huge query, utilising 10 subqueries. This approach failed because it was too slow. When left to run overnight this query did not finish. This query is given in Appendix 1.

Another approach would be to use a package in R called ‘zoo’ for generating a time-series from a set of observations. From here a linear model could be generated using the ‘dynlm’ package which could internally reference the lagged levels and rainfalls given the time series. This approach failed in the testing phase because of a known issue with the ‘dynlm’ package.

A third approach to accessing past values of the time series is to use the ‘astsa’ package in R. This package has a function *lag( )* that allows you to reference the level a specified number of days in the past. However the function used for converting the data into a time series does not support heterogeneous time intervals. Hence the ‘date’ column of the data was effectively ignored, assuming that all observations happened on consecutive days. For the vast majority of the data this is a valid assumption, since data was collected daily and has very few gaps.

However there are some gauges which have a large missing chunk of data in the middle of their history. In these gauges there will be about 10 data points used that are erroneous, the 10 days just after the data collection restarts will be basing predictions off the days just before the gauge stopped. This is not a big deal because it is such a small proportion of the total dataset.

Another hurdle with the time-series implementation is that a time-series can only have a single row of data for each time. In this case we have another dimension to the data which is the guage_id. For a given date there may be many observations taken at different gauges. To manipulate the data into the required format, each gauge was considered separately and a different time series was constructed for each.

The data available within each time series is the amount of rainfall and the river level. These are both continuous numeric data types, and thus a numeric modelling technique is required. The only numeric modelling technique we have learnt about is linear regression, so we will use linear regression to generate the models. A different model will be generated for each gauge.

Modelling Assumptions

In order to do a linear regression on a time series you need to have enough data in each time series. If you try to do a linear regression using the last 10 days history, but you only have 8 days worth of data it will crash. As such we did not model any gauges that had less than 100 days worth of data, to be safe. There definitely would not have been enough data to generate a good model anyway.

Linear regressions in general make 3 major assumptions:

1. Homoscedasticity - the variance of residuals is the same at all x-values
2. The residuals are normally distributed
3. Observations are independent (outside the dependence that is modelled)

For most gauges, the data probably satisfies the first assumption of homoscedasticity since the character of the river is not expected to change very much over time.

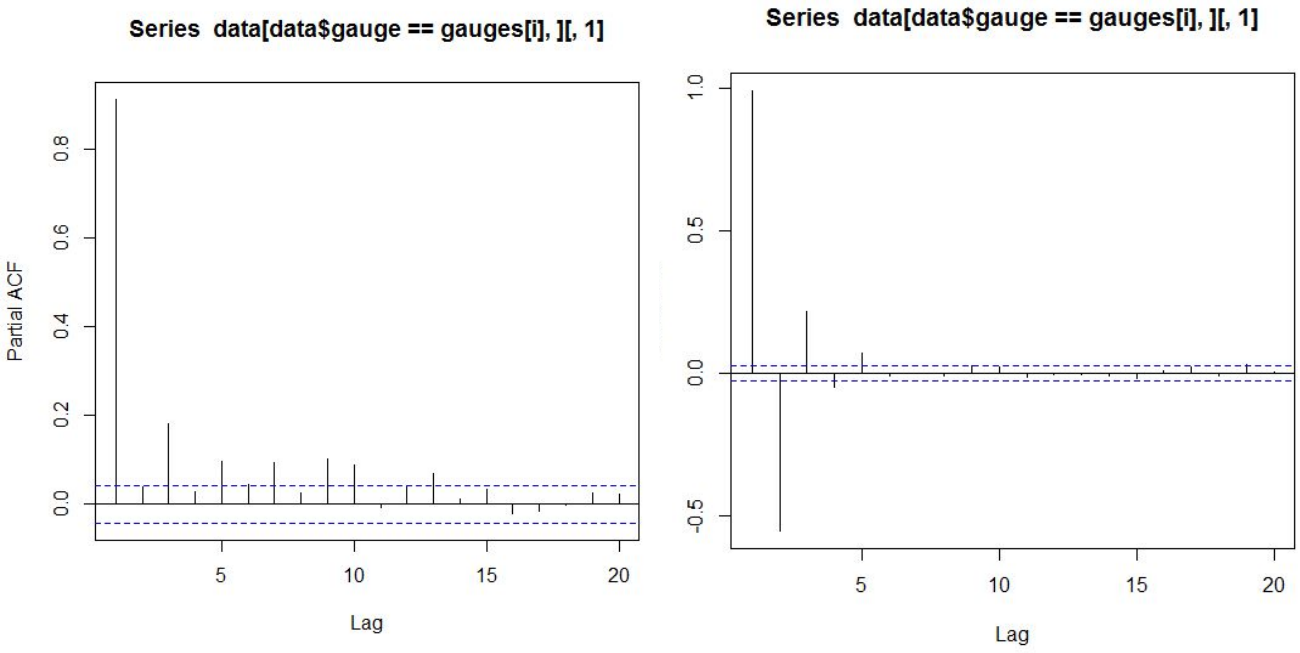
The prediction of rivers will definitely not have normally distributed residuals. There will be much larger errors where the level is high and the prediction is too low than where the level is low and the prediction is too high.

Fortunately linear regression is quite robust to violation of the first 2 assumptions, however it is not so robust to violation of the third assumption.

The third assumption is that the predicted level is independent of any levels or rainfall in the dataset that are not modelled. This means that if we are using 10 days worth of history to make the prediction then the level and rainfall 11 days ago should be independent of the current level.

Intuitively this will be true so long as you use a large enough history to make the predictions. This assumption can be tested by generating a partial autocorrelogram.

Below shows 2 typical partial autocorrelograms for different gauges. In the first example it is very clear that observations beyond 5 days in the past are independent. In the second example this is less clear.



Partial correlogram showing the extra 'information' gained by considering each subsequent lag on two separate gauges.

Generating a Test Design

As described earlier, the data will be separated into training and validation sets. The training set will be the earliest 66% of the data on each gauge, and the validation set is the most recent 33%. This is easily implemented in R using the `head()` and `tail()` functions.

The typical measure of the goodness of a linear regression is the R^2 of the predictions on the validation set. R^2 is a measure of the proportion of explained variability relative to the total variability in the data set. $R^2 = explained\ variability\ /\ total\ variability$ This is a good measure for rivers that have plenty of variability, but it may be deceptively low if the river has very little variability. As an extreme example, if a river is a constant level all the time, and the prediction is just a little bit offset from the actual level all the time then the R^2 will be very low. However, for my purposes this actually a very good prediction. As such we also need another measure of

goodness of fit. #eg. 420003 has an R2 of 0.17048693, RMSE of 0.02248023. This is probs still good.

A more intuitive measure of goodness is the actual amount of error in each prediction. There are several ways of measuring this, such as mean squared error, average absolute dispersion or median absolute deviation . For this application we do not want the measure of goodness to be heavily influenced by outliers. This is because the models will always be bad at predicting very rare events such as floods, and we don't care about predicting the levels during floods accurately. The mean squared error is particularly bad in this regard as it weights larger errors more heavily. We will use the median absolute deviation (MAD) as it is a very robust measure which is not heavily influenced by outliers. This measure also has the benefit of being very intuitive to understand because it simply tells us the median value of the error between the prediction and the actual value. This can be interpreted as a river level (in meters) of how wrong you can expect a 'typical' prediction to be.

A prediction can be considered 'good' if it is has a high R^2 or it has a low MAD. That is, the model is good if it explains most of the variance in the data, OR if it has a small median error.

The gauges need to be predicted at least 2 days in advance, although we will try for 3 days as that would be even better if it works. As such each gauge will have 3 associated models, one for predicting tomorrow, one for predicting 2 days in future, and one for predicting 3 days in the future. Every one of these models can be tested in the same way using R^2 and MAD.

It was determined in the business objective that models not only need to predict the levels well, but they also need to be better than just guessing. A simple guess would be to assume that the river level will not change at all. The R^2 and MAD of a good model should be better than those of this simple guessing model.

Building the models

Parameter Settings

The parameters varied between each of the models were:

- Number of days level lag considered (rainfall is always used up until the day being predicted because accurate rainfall predictions are already available from the BOM)
- Number of days in advance being forecast
- Gauge being predicted (and hence number of observations used to produce the model)

As expected it was found that using more days of history to make the prediction made for slightly better models on the whole. As predicted by the correlograms it was found that each subsequent day added had a diminishing effect on the accuracy of the model.

It was found that predicting more days in advance produced worse predictions. This makes sense since there is less relevant information available.

Model Descriptions

Linear Regression

All the models were generated using the lm() function in R. The predictors were given as a linear combination of lags on rainfall and river levels.

Eg.

```
alldata=ts.intersect(soi,rec,reclag1=lag(rec,-1), reclag2=lag(rec,-2),
      reclag3=lag(rec,-3), soilag1=lag(soi,-1), soilag2=lag(soi,-2),
      soilag3=lag(soi,-3))

Model = lm(soi~soilag1+soilag2+soilag3+rec+reclag1+reclag2+reclag3, data = alldata)
```

	predR2	mad	nrow	gauge	model	daysPred	daysUsed	r2improv	madimprov
reclag5	0.74072200	0.018347009	4472	201001	-0.000892796572638155	1	5	1.747818e-01	0.0005558095
368	0.72406353	0.029228029	4472	201001	c(0.0845167793383726, 0.0376350081891977, 0.44759997...	2	5	4.079167e-01	-0.0033891712
724	0.71393518	0.038214615	4472	201001	c(0.109015880979987, -0.00373167004980997, 0.0876825...	3	5	4.851498e-01	-0.0062633849
1081	0.72216549	0.031673952	4472	201001	c(0.0910485542620964, 0.0605482047395623, 0.45963698...	2	4	4.060187e-01	-0.0009432476
1437	0.73929799	0.019617045	4472	201001	c(0.0565307281060313, 0.622549474566332, 0.066733372...	1	4	1.733578e-01	0.0018258452
1793	0.70979016	0.042338741	4472	201001	c(0.119808270387463, -0.0071795793142228, 0.09893197...	3	4	4.810048e-01	-0.0021392592
2149	0.70314625	0.047594457	4472	201001	c(0.133915829233416, 0.0415461919959262, 0.111333034...	3	3	4.743608e-01	0.0031164574
2505	0.71779875	0.035444753	4472	201001	c(0.101876224770613, 0.0779819784132947, 0.491315806...	2	3	4.016519e-01	0.0028275534
2861	0.73735230	0.021774197	4472	201001	c(0.060824717643706, 0.63896208588793, 0.07060530197...	1	3	1.714121e-01	0.0039829969
13	0.73631444	0.025550830	3797	201900	c(0.259075564501907, 0.638663922125912, -0.005057076...	1	5	2.762904e-01	0.0107248305
369	0.74278030	0.040049865	3797	201900	c(0.428664825952241, 0.0575434810581529, 0.404028387...	2	5	5.258153e-01	0.0133630647
725	0.72621481	0.049395854	3797	201900	c(0.532500714193881, -0.016917032943205, 0.084412253...	3	5	5.832053e-01	0.0123308543
1082	0.74279378	0.040453024	3797	201900	c(0.440963387216979, 0.057570238003081, 0.4032477807...	2	4	5.258288e-01	0.0137662237
1438	0.73639129	0.026828116	3797	201900	c(0.263031715258303, 0.640128639416101, -0.001495555...	1	4	2.763672e-01	0.0120021155
1794	0.72486917	0.051428711	3797	201900	c(0.560292884660554, -0.0024941329156803, 0.07505993...	3	4	5.818597e-01	0.0143637109
2150	0.71775004	0.056555949	3797	201900	c(0.61897628069616, -0.0301433370946541, 0.117583494...	3	3	5.747405e-01	0.0194909491
2506	0.73877447	0.045872336	3797	201900	c(0.485094754059022, 0.0485049085880435, 0.433647362...	2	3	5.218094e-01	0.0191855360

3205 models were generated over the 377 gauges, predicting 1-3 days in advance and using 3-5 days worth of historical data to make that prediction.

Guessing Model

The other model generated was a very simple model which predicts that the river level will not change. This was used as a baseline model to compare the linear regression models. If the linear regression produces worse results than the simple model then it is not a good model at all.

Clustering

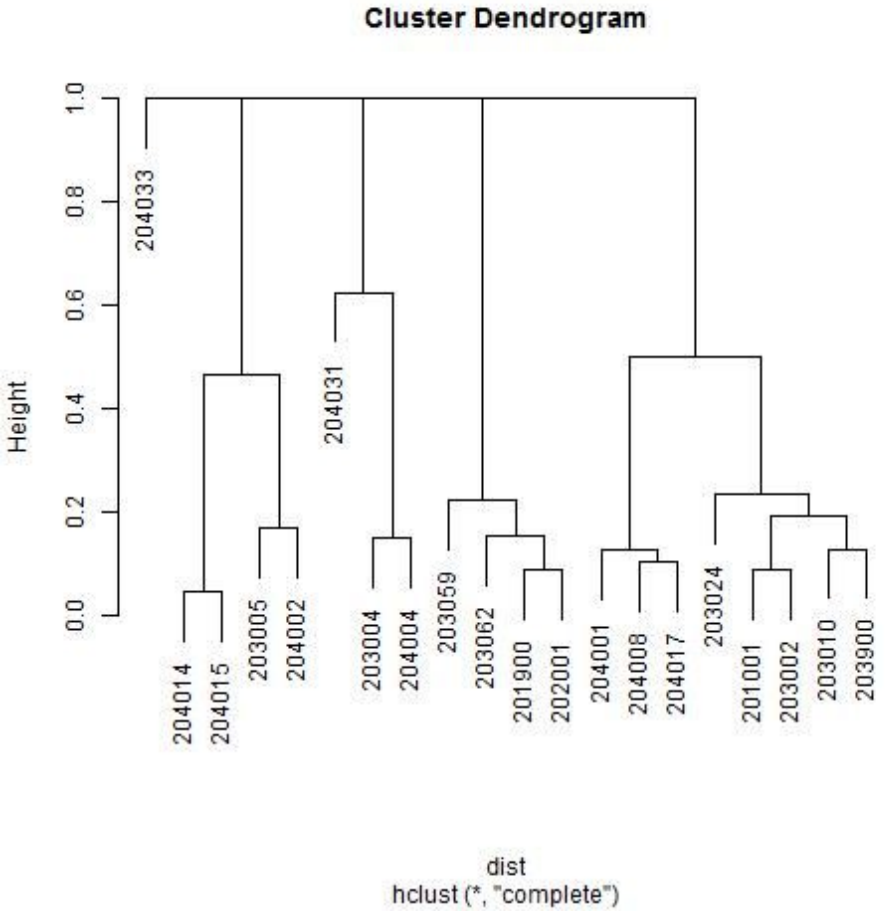
In addition to the linear regression and guessing model, an attempt was made to cluster the rivers and find guages which are very similar to one another. The purpose of this clustering is to be able to predict rivers which have too little data to be able to create an accurate model alone.

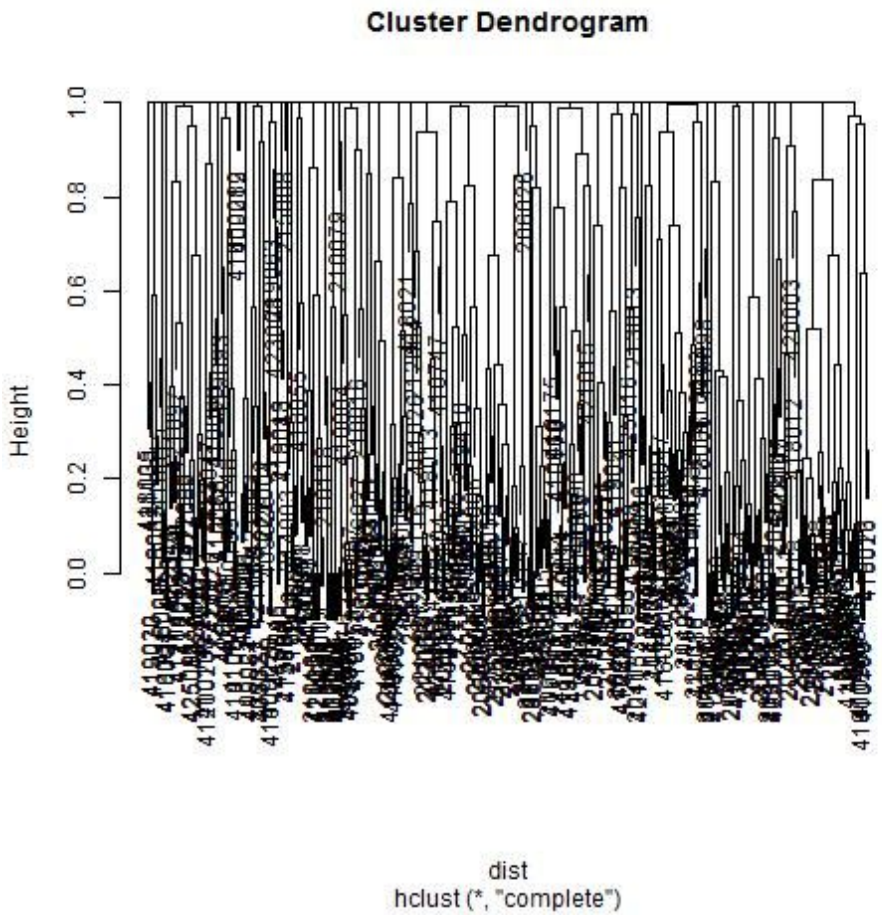
This clustering was tricky because I wanted to cluster based off the river's character, as in its rate of rise and fall and response to rain. However these factors are not independent variables that could easily be passed to a clustering algorithm. Instead I normalised all the river levels and then created my own distance matrix based off the correlation between each pair of rivers. I defined the distance between two rivers to be $1 - |correlation(guageA, gaugeB)|$ because the distance should be positive and greater for pairs of rivers that are less correlated. I then clustered the gauges using this distance (correlation) matrix using a hierarchical clustering algorithm.

Hierarchical clustering is the only type that I could implement with my time-series data because it allowed me to create my own distance matrix. Other clustering techniques such as k-means would not allow me to use my own distance matrix.

If the clusters are cut at a height of 0.1 it would mean that all the rivers within a cluster have a correlation to one another of at least 0.9. This could be used to predict the levels of some rivers which have insufficient data to produce accurate models alone.

This clustering was not eventually implemented because it was too difficult to convert to and from time series data. If I had more time this would be an area to investigate more thoroughly.


Cluster Dendrogram
Height
0.0
0.2
0.4
0.6
0.8
1.0
204033
204031
204014
204015
203005
204002
203004
204004
203059
203062
201900
202001
204001
204008
204017
203024
201001
203002
203010
203900
dist
hclust (*, "complete")



Model Results

The results of each model that were generated are

- R^2
- MAD
- R^2 improvement over simple model for the same gauge
- MAD improvement over simple model for the same gauge
- Model coefficients

Based on the business understanding, the predictions need to be better than a guess in order to be useful. This was quantified as and improvement in R2 and MAD, which was taken to be the difference between the model and the guess. Both values are positive if the model is better than the guess.

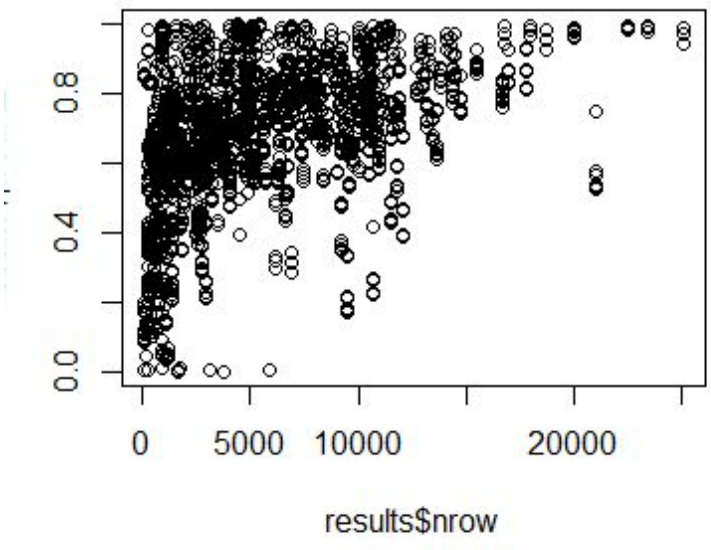
Assessing the models

Revised Parameter Settings

The linear regression produced very good results on some gauges and very poor results on other gauges. The best model were produced when predicting fewer days in advance using more days of historical data. This is shown in the histograms of R^2 and MAD measures in Appendix 2. As such the models used are all those which use the largest history of information (5 days past).

Model Assessment

Some rivers could not be predicted well because there was insufficient data (likely lost in the data cleaning error described earlier). It is clear in the graph below that gauges with less than 5000 observations struggled to make models with a high R^2. Similarly guages with more than 15000 observations all had quite high R^2.



Evaluation

Evaluate Results

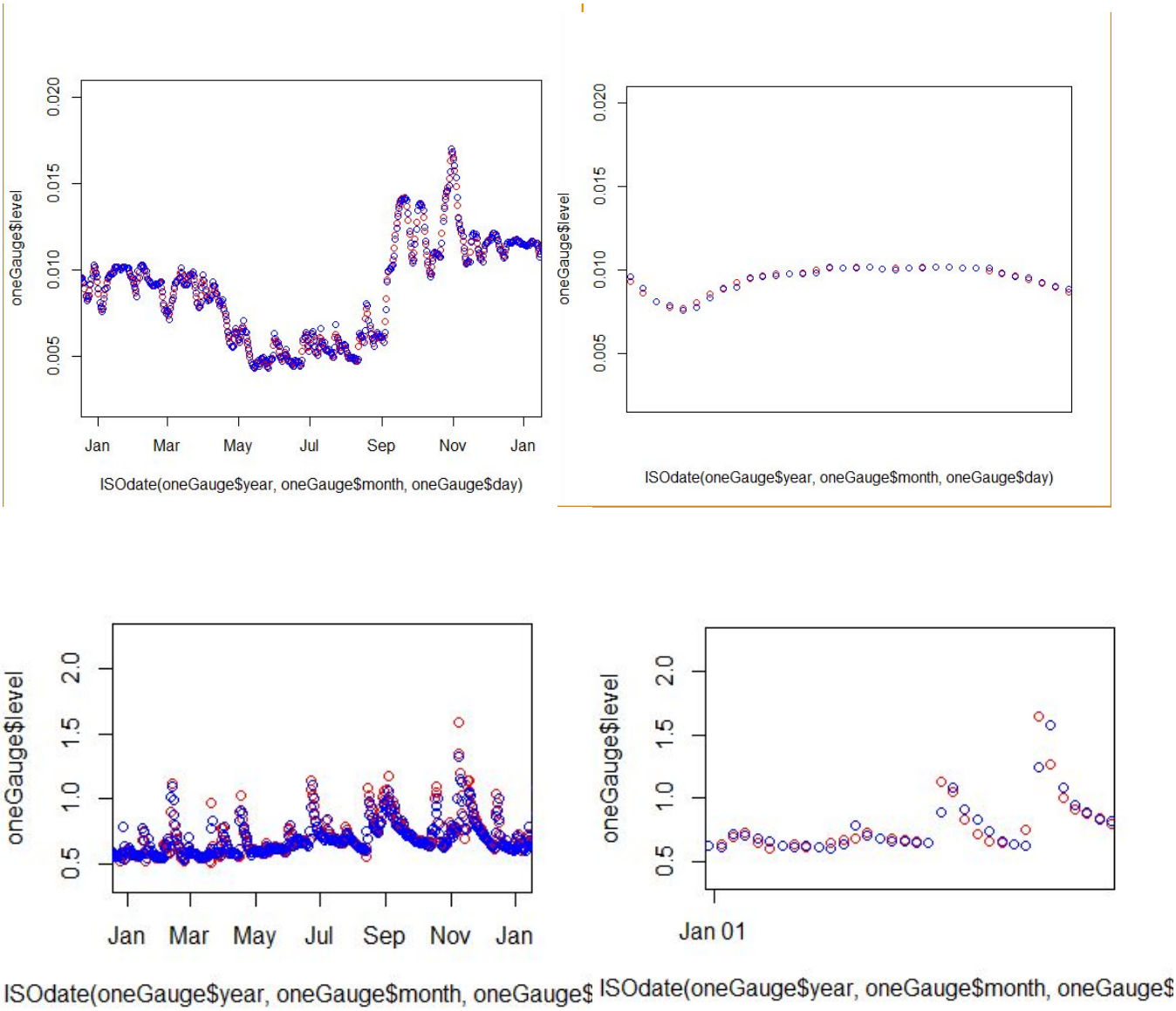
The business understanding required that at least 20% of rivers are predicted accurately 2 days in advance. The business understanding defined acurate as being 50% better than a guess as

to whether or not the gauge will be above its threshold. This measure is not feasible since threshold levels are not easily available for most rivers in NSW. Instead acurate has been redefined as $R^2>0.9$ or MAD<0.05, and both measures are better than that of the guessing model. The choice of 0.05 was chosen because it means that the median error in the prediction of a gauge is 5 cm, and this is a reasonable margin of error that has little influence on how feasible it is to kayak that section. The choice of 0.9 was arbitrarily ‘good’.

The number of rivers predicted well for each forecast period are shown in the table below.

Days forecast	Number of rivers accurately predicted	Portion of total rivers predicted that were accurate
1	257	68.4%
2	187	49.7%
3	136	36.2%

Some rivers were predicted very accurately, for example the ‘MURRAY RIVER AT COROWA’ (409002) is predicted 1 day in advance with $R^2 = 0.99$. And ‘BLUE GUM CREEK AT D/S THIRLMERE LAKES’ (212064) is predicted 1 day in advance with a MAD of 0.0001013307.



These plots show the actual gauge levels (blue) and the predicted gauge levels (red) on 2 rivers (one above, a different river below). The left plots are shown over a time period of a single year, and the right plots are over a month. The above gauge was specifically chosen because the predictions are good, the below guage is more typical. As you can see in the month span, the upper gauge level changes slowly, which is probably why it can be predicted so well. In contrast the lower guage goes from a steady minimum level to a peak in the span of 2 days. You can see the delay between the predicted spike and the true spike.

In general, the rivers that are predicted very well are those that change level slowly. This would suggest that if we had data measured more frequently than daily (say hourly) then we could better predict the rapidly changing rivers.

The rivers that are predicted poorly change level very rapidly. This is possibly due to the inherent delay of a prediction. As seen in the plots above, the lower gauge predicts the peaks quite accurately a day late. Unfortunately, this is not useful information to a kayaker, it is important to know exactly which date the peak will occur on.

Some river appear to be uncorrelated with rainfall, these might be rivers that are close to the ocean and respond to tidal changes, or they might be below dams and respond to dam releases.

In general this collection of models satisfies the business objectives quite well. Although not all rivers were predicted well, many are good enough to be useful. These models will allow the level on some gauges to predicted accurately several days in advance, and hence it will be useful to PaddlersPredictions.

Surprisingly, the rainfall data had very little impact on the predictions. The coefficients for rainfall were in the order of 1000-10000 times smaller than those for the history of river levels. This is potentially because using the rainfall station that is closest to the gauge is not a good estimator. What you really need is the rainfall in the catchment above the gauge.

Approved Models

The approved linear regression models used the full history of 5 days worth of data and are those which satisfy the criteria $R^2>0.9$ or $MAD<0.05$, and both measures are better than that of the guessing model. The exact coefficients of each model are given in the results.rda file included.

Review of Process

Data Understanding Phase

This phase was successful in its purpose of retrieving a sufficient amount of suitable data. It was streamlined by using a webcrawler to download all the data and so if it needed to be done again it could be done so with very little human labour, however it will always take a long time to just download all the files. The main mistake in this phase was not collecting enough data about the thresholds at which each gauge becomes paddleable. This meant that in later stages categorical modelling methods could not be used because there were no meaningful thresholds. One dead end that was mistakenly followed in this stage was politely asking the BOM to send me all of their data in a single file. It is noted that you should not rely on the help of other people or organisations that have no vested interest. An alternative strategy in this stage would have been to keep and make use of the rainfall measurements taken at the gauges. This would have removed the problem of matching rainfall stations to gauges.

Data Preparation Phase

The output from this phase was a database of clean but not easily accessible data. In retrospect this phase should have been approached differently. It was unrealistically optimistic to think that I would be able to create a model that discovers and makes use of all the relevant rainfall stations. It would have been more useful to immediately reduce the size of the dataset to something more manageable by picking it corresponding station to each gauge and discarding all the remaining data. This could have saved a lot of time trying to implement and ODBC in R or write an efficient query to normalise all the river levels and rainfall by gauge in SQL.

The major blunder of this entire project was when I revisited the data preparation phase after abandoning the R-ODBC approach. I did an inner join on the rainfall and river levels tables by date and vicinity. I should have realised at the time, but I missed the fact that if the nearest station to a gauge only had a handful of observations, or its observations were made at a different time to the gauge observations then all that data would be lost. I should have written a query which joined them on the nearest station which had data at that date. This mistaken was overlooked until it was too late to fix.

In future I would not bother with importing all the data into a database at all. I would simply extract the metadata from each file and then decide which files to keep and discard, and create a single csv file from that data. This would save a huge amount of work and make it easier to manipulate the data in R in a reproducible way.

Modelling Phase

This phase produced a set of 580 accurate models. Each model is specific to a single gauge and rainfall station combination.

The biggest time sinks were looking for ways to deal with the time series and get points according to their lag. Similarly to the issues in the data preparation phase, I never should have used a database, it is just too slow. I spent a lot of time trying to create a query that would get the river levels and rainfalls at all the lags but in the end I couldn't use that query because it was too slow. In retrospect I would spend more time learning to use the time series functionalities in R, or I would use a better program like STATA that deals with time series gracefully.

An unresolved issue that I encountered was creating any other type of model (other than a linear regression) using these time series lags. The lags are in a strange data structure that I couldn't easily convert into other structures and other functions did not take input from time series. As such I only created linear regression models and could not use naive bayes or decision tree models. It would have been good to try more different types of models.

Another thing that could be done in the future is to use the clustering to make more effective models of the gauges which don't have enough data. This process failed because a time series cannot be generated from a sequence where there are multiple values for the same date. In retrospect, maybe this issue could be circumvented by normalising the levels and rainfall and then creating a single model on the gauge with the most data and blindly applying it to gauges that are very similar.

Next steps

The next step of this process is deployment. If there were time I would go back and refine the models and try more different variations, but this course is time limited so we must move on.

Possible actions / decisions:

- Implement all the models and provide the user with degrees of certainty about each
- Implement only the accurate models and discard the inaccurate ones
- Investigate the effect of non-linear terms in the linear regression. Eg. rainfall 2days ago * level 3 days ago.
- Go back and use clustering to predict gauges with insufficient data that are strongly correlated to a gauge which can be predicted well
- Go back to the data preparation phase and avoid losing data by linking each gauge to a nearby station which has enough data at the right times
- Go back to the data understanding phase and get data about catchment areas so that rainfall data can be used more effectively.
- Go back to the data collection phase and collect levels data hourly instead of daily.

Appendix 1 - Failed SQL query:

```
"SELECT level, rainfall,
  (
    SELECT K.level
    FROM Levels as K
    WHERE L.gauge_id = K.gauge_id
    AND
    julianday(L.year || '-' || substr('00'||L.month,-2) || '-' ||substr('00'||L.day,-2)) -
    julianday(K.year || '-' || substr('00'||K.month,-2) || '-' ||substr('00'||K.day,-2))
    = -1
    LIMIT 1
  ) as L1,

  (
  SELECT K.level
  FROM Levels as K
  WHERE L.gauge_id = K.gauge_id
  AND
  julianday(L.year || '-' || substr('00'||L.month,-2) || '-' ||substr('00'||L.day,-2)) -
  julianday(K.year || '-' || substr('00'||K.month,-2) || '-' ||substr('00'||K.day,-2))
  = -2
  LIMIT 1
  ) as L2,

  (
  SELECT K.level
  FROM Levels as K
  WHERE L.gauge_id = K.gauge_id
  AND
  julianday(L.year || '-' || substr('00'||L.month,-2) || '-' ||substr('00'||L.day,-2)) -
  julianday(K.year || '-' || substr('00'||K.month,-2) || '-' ||substr('00'||K.day,-2))
  = -3
  LIMIT 1
  ) as L3,

  (
  SELECT K.level
  FROM Levels as K
  WHERE L.gauge_id = K.gauge_id
  AND
  julianday(L.year || '-' || substr('00'||L.month,-2) || '-' ||substr('00'||L.day,-2)) -
  julianday(K.year || '-' || substr('00'||K.month,-2) || '-' ||substr('00'||K.day,-2))
  = -4
  LIMIT 1
  ) as L4,

  (
  SELECT K.level
```

```
 FROM Levels as K
 WHERE L.gauge_id = K.gauge_id
 AND
 julianday(L.year || '-' || substr('00'||L.month,-2) || '-' ||substr('00'||L.day,-2)) -
 julianday(K.year || '-' || substr('00'||K.month,-2) || '-' ||substr('00'||K.day,-2))
 = -5
 LIMIT 1
 ) as L5,

(
 SELECT K.Rainfall
  FROM Rainfall as K
  WHERE R.station_id = K.station_id
  AND
  julianday(R.year || '-' || substr('00'||R.month,-2) || '-' ||substr('00'||R.day,-2)) -
  julianday(K.year || '-' || substr('00'||K.month,-2) || '-' ||substr('00'||K.day,-2))
  = -1
  LIMIT 1
) as R1,

(
 SELECT K.Rainfall
 FROM Rainfall as K
 WHERE R.station_id = K.station_id
 AND
 julianday(R.year || '-' || substr('00'||R.month,-2) || '-' ||substr('00'||R.day,-2)) -
 julianday(K.year || '-' || substr('00'||K.month,-2) || '-' ||substr('00'||K.day,-2))
 = -2
 LIMIT 1
) as R2,

(
 SELECT K.Rainfall
 FROM Rainfall as K
 WHERE R.station_id = K.station_id
 AND
 julianday(R.year || '-' || substr('00'||R.month,-2) || '-' ||substr('00'||R.day,-2)) -
 julianday(K.year || '-' || substr('00'||K.month,-2) || '-' ||substr('00'||K.day,-2))
 = -3
 LIMIT 1
) as R3,

(
 SELECT K.Rainfall
 FROM Rainfall as K
 WHERE R.station_id = K.station_id
 AND
 julianday(R.year || '-' || substr('00'||R.month,-2) || '-' ||substr('00'||R.day,-2)) -
 julianday(K.year || '-' || substr('00'||K.month,-2) || '-' ||substr('00'||K.day,-2))
 = -4
```

```
 LIMIT 1
) as R4,

(
 SELECT K.Rainfall
 FROM Rainfall as K
 WHERE R.station_id = K.station_id
 AND
 julianday(R.year || '-' || substr('00'||R.month,-2) || '-' ||substr('00'||R.day,-2)) -
 julianday(K.year || '-' || substr('00'||K.month,-2) || '-' ||substr('00'||K.day,-2))
 = -5
 LIMIT 1
) as R5



        FROM Levels as L JOIN Gauges as G ON (L.gauge_id = G.id)
        JOIN Rainfall as R ON (R.station_id = G.nearestStat)
        WHERE L.year=R.year AND L.month=R.month AND L.day=R.day
        "
```

Appendix 2 - Histograms of error measurements

These histograms show the distribution of MAD and R^2 across all rivers when the number of days used are varied. You can see that as the number of days used increases, both the R2 increases and the MAD decreases. Although, not by very much.