Data preparation

Willem Vervoort, Michaela Dolk & Floris van Ogtrop

2018-08-31

```
require(zoo)
require(pander)
require(reshape2)
require(xts)
require(curl)
require(tidyverse)
require(lubridate)
panderOptions('table.split.cells', 12)
panderOptions('table.alignment.default', 'center')
panderOptions('table.alignment.rownames', 'right')
```

```
# root dir
knitr::opts_knit$set(root.dir = "C:/Users/rver4657/ownCloud/Virtual Experiments/VirtExp")
```

This rmarkdown document and the resulting pdf are stored on github. All directories (apart from the root working directory) refer to the directories in this repository

Introduction

This document is related to the manuscript "Disentangling climate change trends in Australian streamflow" (vervoort et al.), submitted to Journal of Hydrology. This document outlines the preparation of the original data into the dataframes that have been analysed in the project. The decision making on how stations were identified, is outlined in the methods of the submitted manuscript. This document is aimed at documenting the code of the analysis.

sources of data

As outlined in the manuscript, the original data were sourced from the following locations: Streamflow is from the Bureau of Meteorology (BOM) hydrological reference stations http://www.bom.gov.au/water/hrs/ Rainfall and temperature station data were obtained from the BOM station data http://www.bom.gov.au/climate/data-services/ Gridded data were obtained from ANUClimate collection in the NCI Catalogue or via ANDS: https://researchdata.ands.org.au/anuclimate-collection/983248

Reading in the data

The data consists of comma delimited (csv) files, as downloaded from the websites. All data cover the period 1970 - 2010. The following flow stations were used:

Table 1: Stations used in this project (continued below)

Name.used.in.this.project	Number	Region	Catchment.area.smaller.250km2.
COTT	410730	ACT	130
RUTH	219001	NSW	14
CORA	215004	NSW	166

Name.used.in.this.project	Number	Region	Catchment.area.smaller.250km2.
ELIZ	G8150018	NT	96
COCH	113004A	QLD	95
COEN	922101B	QLD	170
SCOT	A5030502	SA	29
HELL	312061	TAS	101
NIVE	304497	TAS	174
MURR	405205	VIC	106
SOUT	225020A	VIC	10
YARR	614044	WA	80
DOMB	607155	WA	116

Latitude	Longitude	Rain.St	HQTmax.st
-35.59	148.8	70316	70351
-36.59	149.4	69003	70351
-35.15	150	69049	68072
-12.61	131.1	14149	14015
-17.74	145.6	31083	34084
-13.96	143.2	27005	27045
-35.1	138.7	23734	23373
-41.42	145.7	96023	96003
-42.03	146.4	91065	96003
-37.41	145.6	88028	85072
-37.83	146.4	85238	85072
-32.81	116.2	9538	9021
-34.58	116	9590	9518

Define decades to analyze

```
study_period_decades <- c("70_80", "80_90", "90_00", "00_10")
decade_start <- c(as.Date("1/1/1970", format="%d/%m/%Y"),
                  as.Date("1/1/1980", format="%d/%m/%Y"),
                  as.Date("1/1/1990", format="%d/%m/%Y"),
                  as.Date("1/1/2000", format="%d/%m/%Y"))
decade_end <- c(as.Date("31/12/1979", format="%d/%m/%Y"),
                as.Date("31/12/1989", format="%d/%m/%Y"),
                as.Date("31/12/1999", format="%d/%m/%Y"),
                as.Date("31/12/2010", format="%d/%m/%Y"))

# define the overall period
start_date <- as.Date("1970-01-01")
end_date <- as.Date("2010-12-31")
```

Read in the daily stream flow data

This includes conversion from ML/day (as indicated on the source website) to mm to match the rainfall data and to use in models. This means that the data needs to be scaled to the catchment size:

- convert ML/day to mm

- 1 ML = 10^6 L = 10^6 dm^3 is 10^9 cm^3 is 10^{12} mm^3
- 1 km^2 = 10^6 m^2 = 10^{10} cm^2 = 10^{12} mm^2
- ML/day to mm –> flow (ML/day) /area(km^2) = mm/day

```
# read in the flow data and convert to zoo
for (i in seq_along(Stations[,1])) {
  temp <- read.csv(paste("data/Original streamflow data/", Stations[i,2],
                         "_daily_ts2.csv", sep=""))
  year <- substr(as.character(temp$Date), nchar(as.character(temp$Date))-1,
                 nchar(as.character(temp$Date)))
  Dates <- as.Date(paste(substr(as.character(temp$Date), 1,
                                nchar(as.character(temp$Date))-2),
                         ifelse(as.numeric(year)>=50,paste("19",year,sep=""),
  assign(paste(Stations[i,1], "_daily_flow", sep=""),
         zoo(temp$Q/(Stations[i,4]),order.by=Dates))
}
  #####

# use zoo to merge all catchments to use same time interval
flow_zoo<-merge(COTT_daily_flow, RUTH_daily_flow, CORA_daily_flow,
                ELIZ_daily_flow, COCH_daily_flow, COEN_daily_flow,
                SCOT_daily_flow, HELL_daily_flow, NIVE_daily_flow,
                MURR_daily_flow, SOUT_daily_flow, YARR_daily_flow,
                DOMB_daily_flow)
# limit to 1970 - 2010
flow_zoo <- window(flow_zoo, start=start_date, end=end_date)

#####
```

Read in the Rainfall stations

This section reads in the data related to the closest possible rainfall stations.

```
closerainfall_stns <- Stations[,7]

# read in the data and subset to the required period
for (i in seq_along(closerainfall_stns)) {
  temp<-read.csv(paste("data/Original Rainfall data/", "IDCJAC0009_",
                       ifelse(nchar(closerainfall_stns[i])<5,
                        "00",
                        ifelse(nchar(closerainfall_stns[i])<6,"0","")),
                       closerainfall_stns[i], "_1800_Data.csv", sep=""))
  temp$Date<-ISOdate(year=temp$Year, month=temp$Month, day=temp$Day)
  temp$Date<-as.Date(temp$Date)
  temp<-subset(temp, Date>=start_date & Date<=end_date,
               select=c(9, 6))
  colnames(temp) <- c("Date", "Rainfall")
  assign(paste(Stations[i,1], "Rain", sep=""),
         zoo(temp$Rainfall, order.by=temp$Date))
}
# merge
rain_zoo<-merge(COTTRain, RUTHRain, CORARain, ELIZRain, COCHRain,
                COENRain, SCOTRain, HELLRain, NIVERain, MURRRain,
                SOUTRain, YARRRain, DOMBRain)
```

Read in the Temperature stations

```
HQmaxT_stns <- Stations[,8]


for (i in seq_along(HQmaxT_stns)) {
  temp <- read.csv(paste("data/Original Temperature data/",
                         ifelse(nchar(HQmaxT_stns[i])<5,
                          "00",
                          ifelse(nchar(HQmaxT_stns[i])<6,"0","")),
                           HQmaxT_stns[i], ".csv", sep=""))
  temp$Date <- as.Date(temp$Date, format="%d/%m/%Y")
  temp$maxT[temp$maxT==99999.9] <- NA
  temp<-subset(temp, Date>=start_date & Date<=end_date)
  assign(paste(Stations[i,1], "temp.maxT", sep=""),
         zoo(temp$maxT, order.by=temp$Date))
}
# merge and create data.frame
maxT_zoo<-merge(COTTtemp.maxT,RUTHtemp.maxT,CORAtemp.maxT,
                ELIZtemp.maxT,COCHtemp.maxT,COENtemp.maxT,
                SCOTtemp.maxT,HELLtemp.maxT,NIVEtemp.maxT,
                MURRtemp.maxT,SOUTtemp.maxT,YARRtemp.maxT,
                DOMBtemp.maxT)
```

download the gridded rainfall data

```
# define parts of the Thredds url links to access the data
# commented out as speeds up compiling, data already downloaded
# part1 <- "http://dapds00.nci.org.au/thredds/ncss/rr9/eMAST_data/ANUClimate/ANUClimate_v1-0_rainfall_d
# part2 <- "&time_start=1970-01-01T00%3A00%3A00Z&time_end=2010-12-31T00%3A00%3A00Z&accept=csv_file"
#
# for (i in 1:nrow(Stations)) {
#   url <- paste(part1,Stations$Latitude[i],"&longitude=",
#                Stations$Longitude[i],part2, sep="")
#
#   curl_download(url,destfile=paste("data/",Stations[i,1],
#                                    "_ANUclimRain.csv",sep=""))
#
# }

# combine ANUclim data together in a data frame similar to flow_zoo
for (i in 1:nrow(Stations)) {
    temp<-read.csv(paste("data/",Stations[i,1],
                         "_ANUclimRain.csv",sep=""))
    temp$time<-ymd(substr(temp$time,1,10))
    temp <- temp %>%
      select(time, `lwe_thickness_of_precipitation_amount.unit.mm.day.1.`)
    colnames(temp) <- c("Date", "Rainfall")
    assign(paste(Stations[i,1], "RainAC", sep=""),
           zoo(temp$Rainfall, order.by=temp$Date))
  }
  # merge
```

```
  rainAC_zoo<-merge(COTTRainAC, RUTHRainAC, CORARainAC, ELIZRainAC, COCHRainAC,
                    COENRainAC, SCOTRainAC, HELLRainAC, NIVERainAC, MURRRainAC,
                    SOUTRainAC, YARRRainAC, DOMBRainAC)
```

Summarising to weekly data

Because all the statistical analyses were run on weekly data, the summaries were all created the same way.

```
flow_weekly <- as.tibble(flow_zoo) %>%
  group_by(year = year(time(flow_zoo)),week = week(time(flow_zoo))) %>%
  summarise_at(vars(COTT_daily_flow:DOMB_daily_flow), sum, na.rm=T)

dates <- ymd(paste(flow_weekly$year,"01","01",sep="-")) +
  weeks(flow_weekly$week)
# remove year and week column
flow_weekly <- flow_weekly %>%
  ungroup() %>%
  select(COTT_daily_flow:DOMB_daily_flow)
# rebuild as a zoo
flow_weekly <- zoo(flow_weekly, order.by = dates)

rain_weekly <- as.tibble(rain_zoo) %>%
  group_by(year = year(time(rain_zoo)),week = week(time(rain_zoo))) %>%
  summarise_at(vars(COTTRain:DOMBRain), sum, na.rm=T)

# remove year and week column
rain_weekly <- rain_weekly %>%
  ungroup() %>%
  select(COTTRain:DOMBRain)
# rebuild as a zoo
rain_weekly <- zoo(rain_weekly, order.by = dates)

maxT_weekly <- as.tibble(maxT_zoo) %>%
  group_by(year = year(time(maxT_zoo)),week = week(time(maxT_zoo))) %>%
  summarise_at(vars(COTTtemp.maxT:DOMBtemp.maxT), sum, na.rm=T)

# remove year and week column
maxT_weekly <- maxT_weekly %>%
  ungroup() %>%
  select(COTTtemp.maxT:DOMBtemp.maxT)
# rebuild as a zoo
maxT_weekly <- zoo(maxT_weekly, order.by = dates)

# gridded rainfall
raingrid_weekly <- as.tibble(rainAC_zoo) %>%
  group_by(year = year(time(rainAC_zoo)),week = week(time(rainAC_zoo))) %>%
  summarise_at(vars(COTTRainAC:DOMBRainAC), sum, na.rm=T)

# remove year and week column
raingrid_weekly <- raingrid_weekly %>%
  ungroup() %>%
  select(COTTRainAC:DOMBRainAC)
# rebuild as a zoo
```

```
raingrid_weekly <- zoo(raingrid_weekly, order.by = dates)
```

Stacking and merging weekly data into one dataset

Now stack all the data together to create one transportable data set

```
# flow
flow_weekly_stack <- data.frame(Date=time(flow_weekly),
                                coredata(flow_weekly))
colnames(flow_weekly_stack) <- c("Date",
                                 paste("flow",Stations[,1],sep="."))

# add a column for the decade
for (j in 1:length(decade_start)) {
  flow_weekly_stack$decade[as.Date(flow_weekly_stack$Date) >=
                             as.Date(decade_start[j]) &
                             as.Date(flow_weekly_stack$Date) <=
                             as.Date(decade_end[j])] <-
    study_period_decades[j]
}
# stack
flow_weekly_stack <- reshape(flow_weekly_stack, direction="long",
                             varying=2:14, sep=".")

# Now do the same for rainfall
rain_weekly_stack <- data.frame(Date=time(rain_weekly),
                                coredata(rain_weekly))
colnames(rain_weekly_stack) <- c("Date",
                                 paste("rain",Stations[,1],sep="."))

# add a column for the decade
for (j in 1:length(decade_start)) {
  rain_weekly_stack$decade[as.Date(rain_weekly_stack$Date) >=
                             as.Date(decade_start[j]) &
                             as.Date(rain_weekly_stack$Date) <=
                             as.Date(decade_end[j])] <-
    study_period_decades[j]
}
# stack
rain_weekly_stack <- reshape(rain_weekly_stack, direction="long",
                             varying=2:14, sep=".")


# Now do the same for raingridfall
raingrid_weekly_stack <- data.frame(Date=time(raingrid_weekly),
                                coredata(raingrid_weekly))
colnames(raingrid_weekly_stack) <- c("Date",
                                 paste("raingrid",Stations[,1],sep="."))

# add a column for the decade
for (j in 1:length(decade_start)) {
  raingrid_weekly_stack$decade[as.Date(raingrid_weekly_stack$Date) >=
                             as.Date(decade_start[j]) &
```

```
                              as.Date(raingrid_weekly_stack$Date) <=
                              as.Date(decade_end[j])] <-
    study_period_decades[j]
}
# stack
raingrid_weekly_stack <- reshape(raingrid_weekly_stack, direction="long",
                              varying=2:14, sep=".")

# and for temperature
maxT_weekly_stack <- data.frame(Date=time(maxT_weekly),
                                coredata(maxT_weekly))
colnames(maxT_weekly_stack) <- c("Date",
                               paste("maxT",Stations[,1],sep="."))

# add a column for the decade
for (j in 1:length(decade_start)) {
  maxT_weekly_stack$decade[as.Date(maxT_weekly_stack$Date) >=
                              as.Date(decade_start[j]) &
                              as.Date(maxT_weekly_stack$Date) <=
                             as.Date(decade_end[j])] <-
    study_period_decades[j]
}
# stack
maxT_weekly_stack <- reshape(maxT_weekly_stack, direction="long",
                              varying=2:14, sep=".")


# Now merge all together into one dataset
flow_rain_maxT_weekly <- cbind(flow_weekly_stack[,1:4],
                               rain_weekly_stack[,4],
                               maxT_weekly_stack[,4],
                               raingrid_weekly_stack[,4])
colnames(flow_rain_maxT_weekly) <- c("Date","Decade", "Station", "Flow",
                                     "Rain", "MaxT", "gridRain")
```

Gridded rainfall comparison

This is a comparison of the gridded ANUCLIM rainfalldata against the station rainfall data.

```
GridRainAllDataout <- as.tibble(rainAC_zoo) %>%
  gather(key= "Station", value="gridRain", COTTRainAC:DOMBRainAC)

GridRainAllDataout$Rain <- melt(as.data.frame(rain_zoo))[,2]
```

```
## No id variables; using all as measure variables
```

```
GridRainAllDataout$Flow <- melt(as.data.frame(flow_zoo))[,2]
```

```
## No id variables; using all as measure variables
```

```
GridRainAllDataout$maxT <- melt(as.data.frame(maxT_zoo))[,2]
```

```
## No id variables; using all as measure variables
```

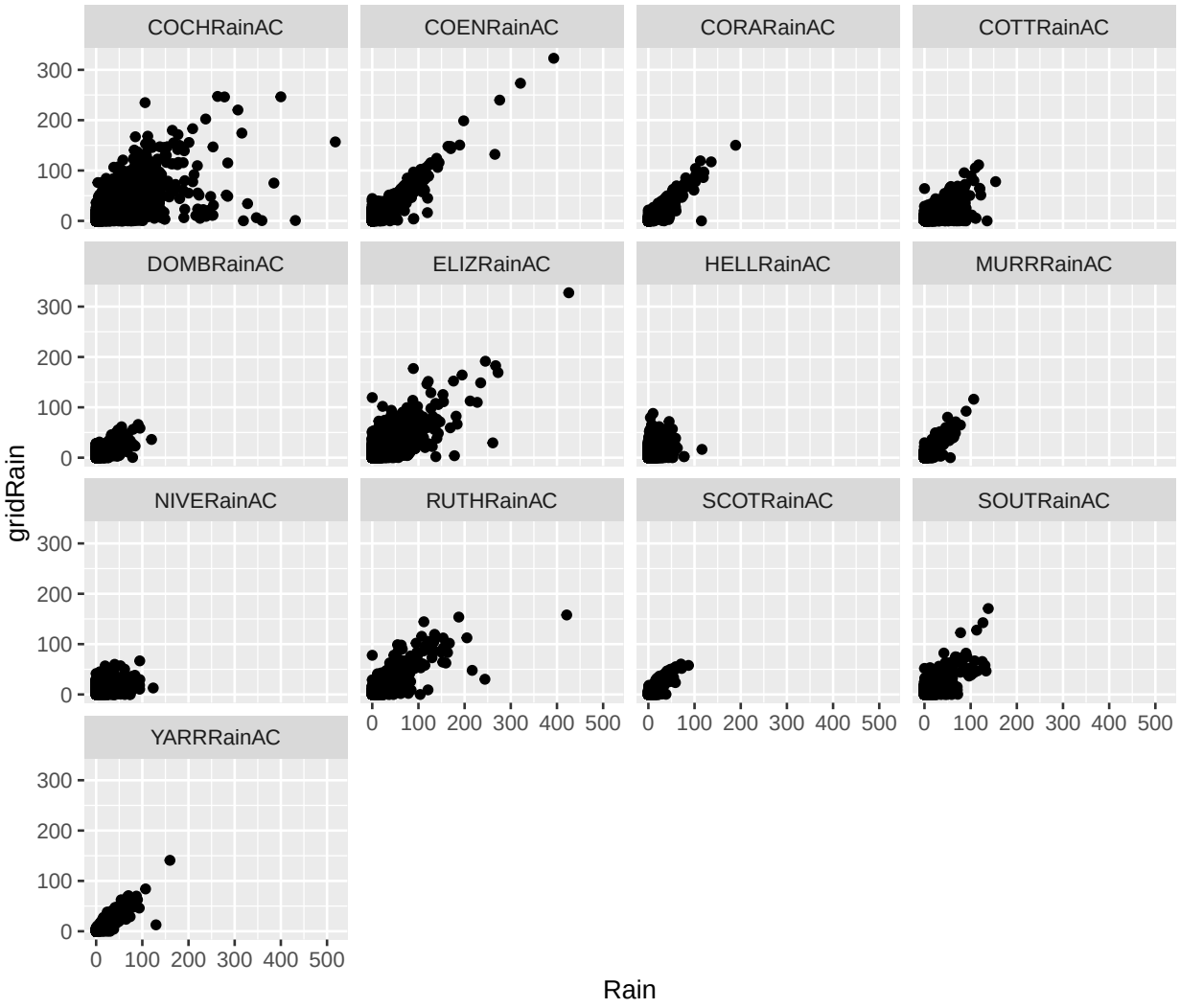

Figure 1: Simple x-y plot comparing the gridded and the rainfall station data

```
# First make a simple xyplot by catchment
xyp_rain <- ggplot(data=GridRainAllDataout,aes(x=Rain,  y=gridRain))
xyp_rain <- xyp_rain + geom_point() + facet_wrap(~Station)
print(xyp_rain)
```

Figure 1 shows reasonable 1:1 agreement between the gridded and the station rainfall, The gridded rainfall is an interpolated surface based on several stations, so not all data should directly match.
To do another comparison, we should also compare the monthly data, just to check that we have actually extracted the right stations.

```
gridRain_m <- as.tibble(rainAC_zoo) %>%
  mutate(year=year(time(rainAC_zoo)), month= month(time(rainAC_zoo))) %>%
  group_by(year, month) %>%
  summarise_at(vars(COTTRainAC:DOMBRainAC), sum,na.rm=T) %>%
  ungroup() %>%
  gather(key = "Station", value= "gridRain", COTTRainAC:DOMBRainAC)

rain_m <- as.tibble(rain_zoo) %>%
  group_by(year=year(time(rain_zoo)), month= month((time(rain_zoo)))) %>%
  summarise_all(sum, na.rm=T) %>%
  ungroup() %>%
  gather(key = "Station", value= "Rain", COTTRain:DOMBRain)

plot_df <- rain_m %>%
  mutate(gridRain = gridRain_m$gridRain) %>%
  ggplot(aes(x=Rain,  y=gridRain)) + geom_point() +
  facet_wrap(~Station)
print(plot_df)
```

That looks a even better, which why we decided to use ANUClim rather than BOM gridded data from 2018

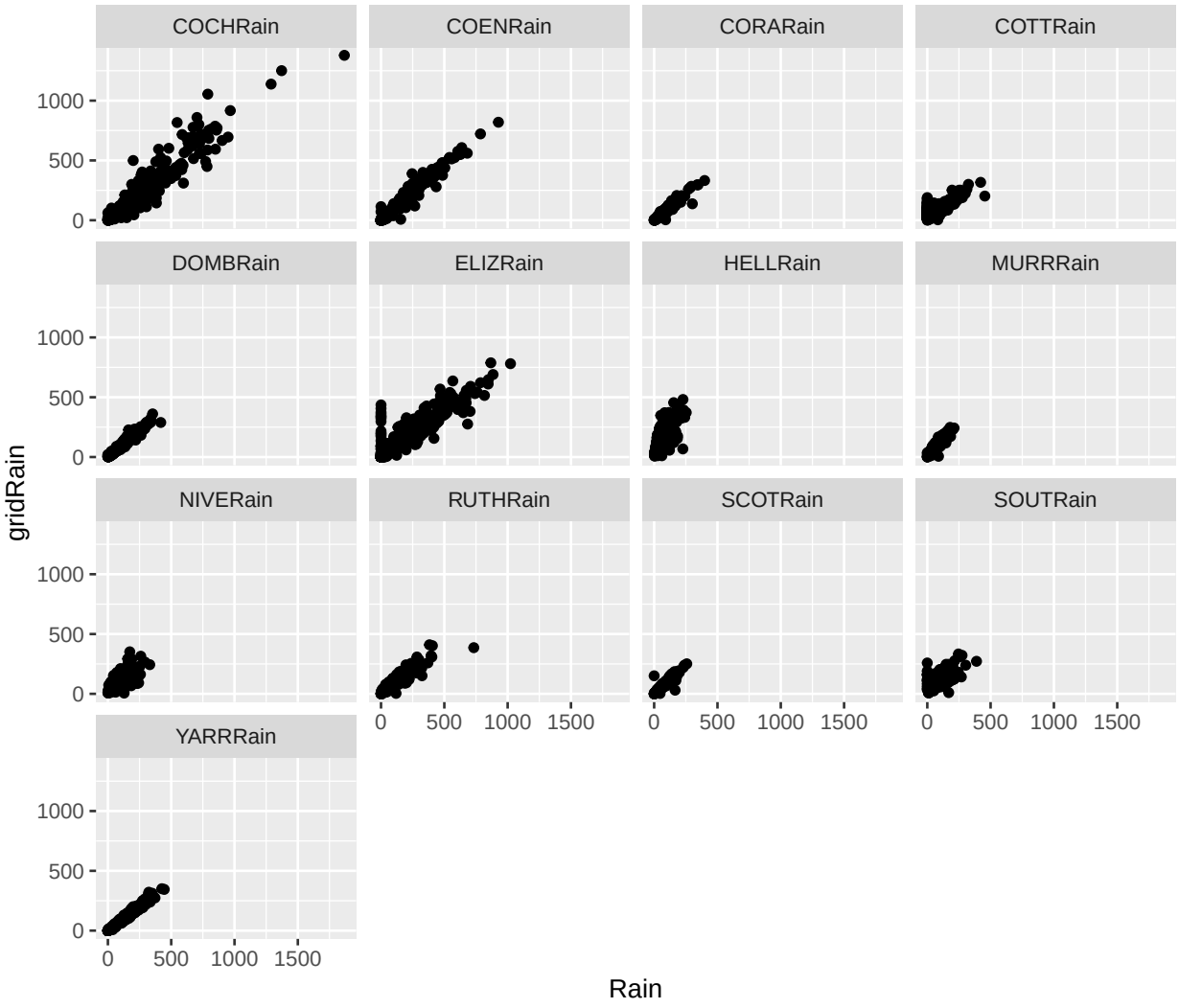

Figure 2: comparison of monthly rainfall data between gridded and station data

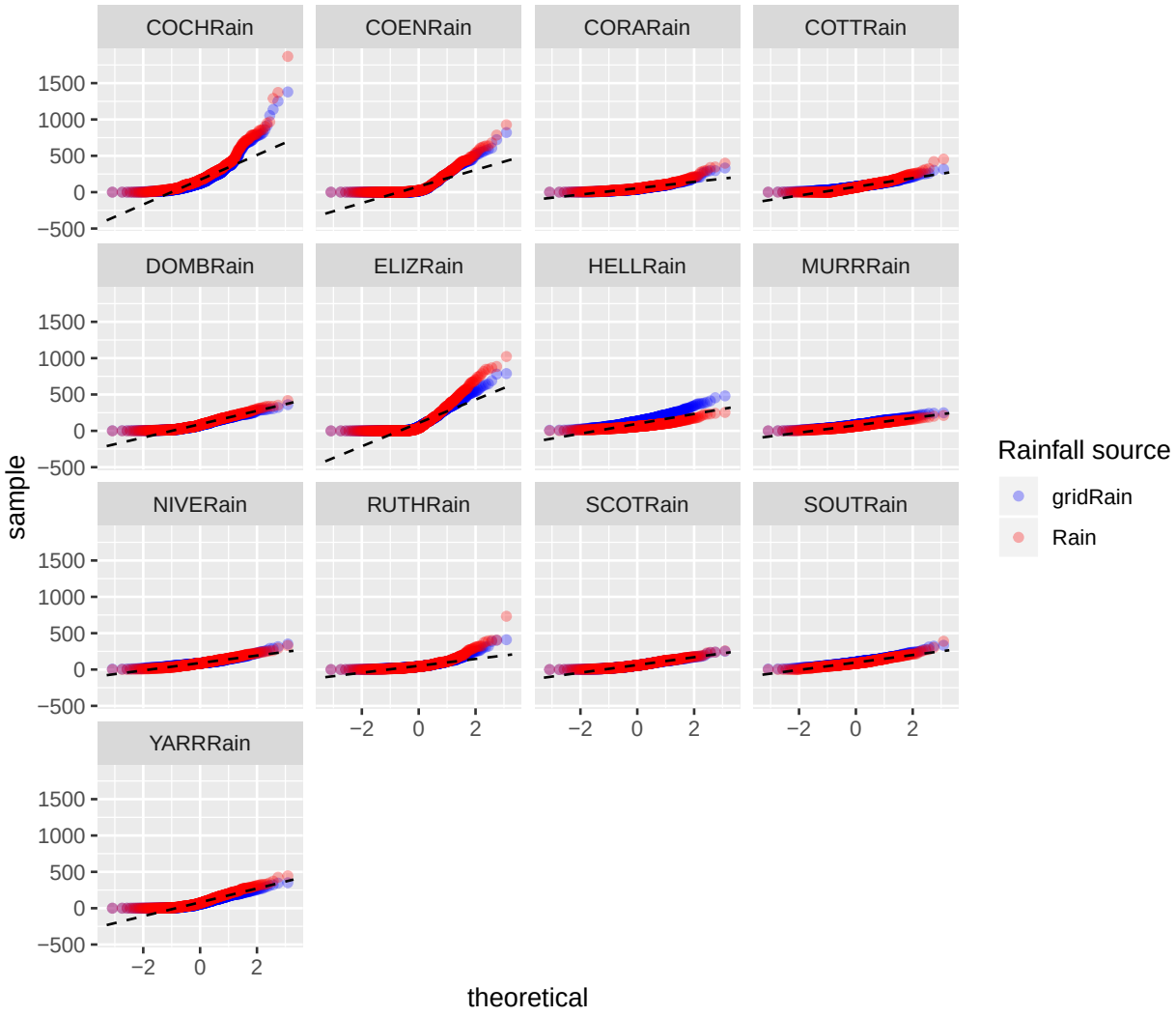

Figure 3: QQplot comparing gridded rainfall and station data

Another way to compare the data is probably using a qqplot.This is a bit trickier, as we need to calculate quantiles We can only plot them against normal quantiles in ggplot()

```
# make a combined data frame
plot_df <- rain_m %>%
  mutate(gridRain = gridRain_m$gridRain) %>%
  # gather rain and gridRain
  gather(key="rain_data_source", value="Rainfall", Rain:gridRain) %>%
  # now plot using geom_qq
  ggplot() + stat_qq(aes(sample=Rainfall, colour=rain_data_source), alpha=0.3) +
  facet_wrap(~Station) + stat_qq_line(aes(sample=Rainfall),linetype="dashed") +
  scale_colour_manual("Rainfall source",
                      values=c("Rain" = "red", "gridRain" = "blue"))
print(plot_df)
```

Figure 3 shows fairly good agreement across the distributions with only the high rainfall quantiles showing some deviations. This is logical as the gridded rainfall is a smoothed replica of the station data due to the

interpolation.

Another important characteristic of rainfall could be the memory,so it might be worth comparing autocorrelation graphs for the different series.

```
# Now show acfs
# Gridded rainfall
gridacf <- tapply(GridRainAllDataout$gridRain,
                            GridRainAllDataout$Station,
                            acf, plot = FALSE)
gridacfdf <- do.call(rbind,lapply(gridacf,function(x)
  with(x,data.frame(lag, acf))))
gridacfdf$Station <- rep(unique(GridRainAllDataout$Station),each=42)


acfGrid <- ggplot(data = gridacfdf, mapping = aes(x = lag, y = acf)) +
  geom_hline(aes(yintercept = 0)) +
  geom_segment(mapping = aes(xend = lag, yend = 0)) +
  facet_wrap(~Station)

# Now normal rainfall
rainacf <- tapply(GridRainAllDataout$Rain,
                  GridRainAllDataout$Station,
                  acf, plot = FALSE,na.action=na.pass)
rainacfdf <- do.call(rbind,lapply(rainacf,function(x)
  with(x,data.frame(lag, acf))))
rainacfdf$Station <- rep(unique(GridRainAllDataout$Station),each=42)


acfRain <- ggplot(data = rainacfdf, mapping = aes(x = lag, y = acf)) +
  geom_hline(aes(yintercept = 0)) +
  geom_segment(mapping = aes(xend = lag, yend = 0)) +
  facet_wrap(~Station)
```

Now print both

```
print(acfGrid)
```

```
print(acfRain)
```

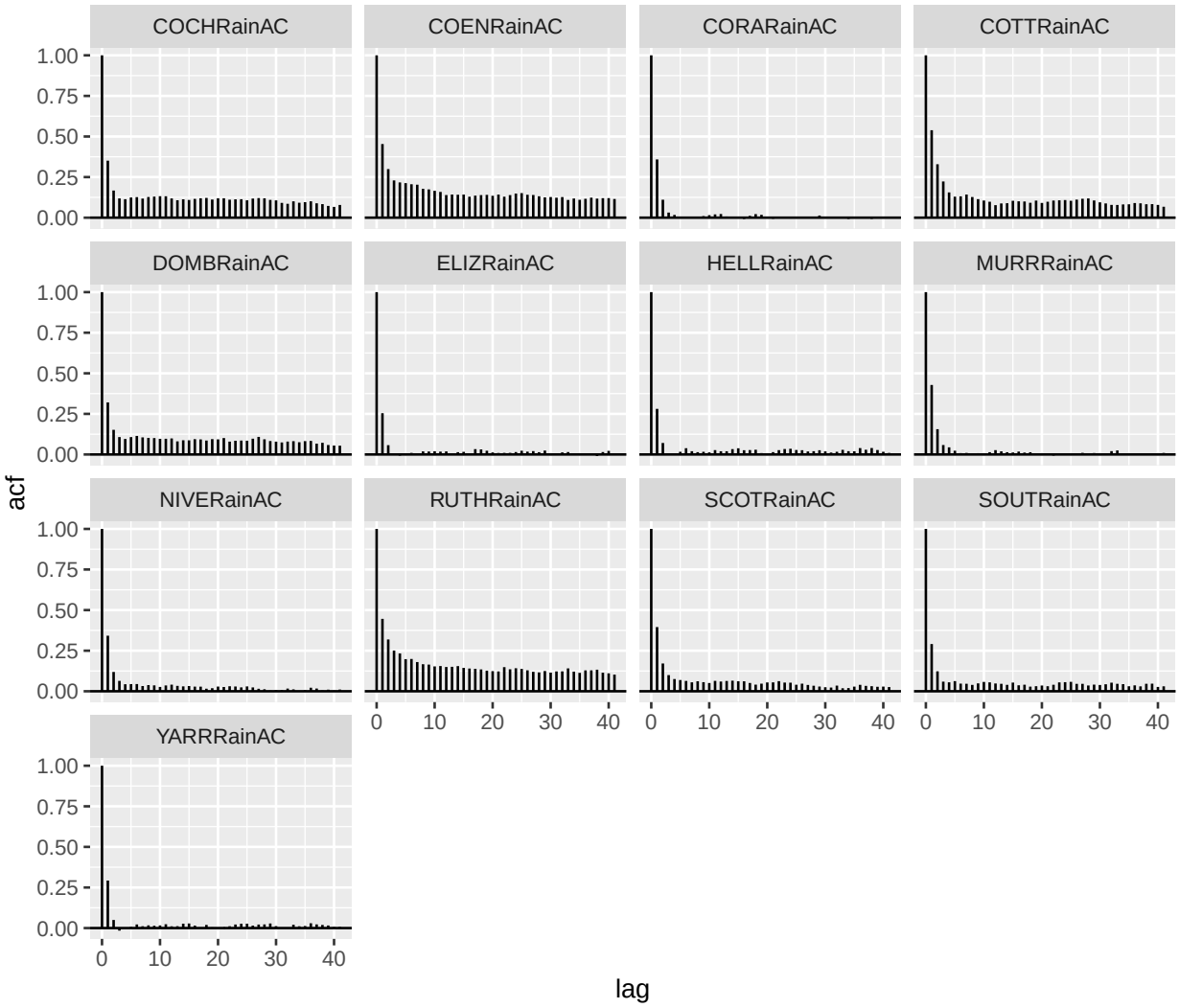

Figure 4: Autocorrelation plot of the gridded rainfall data

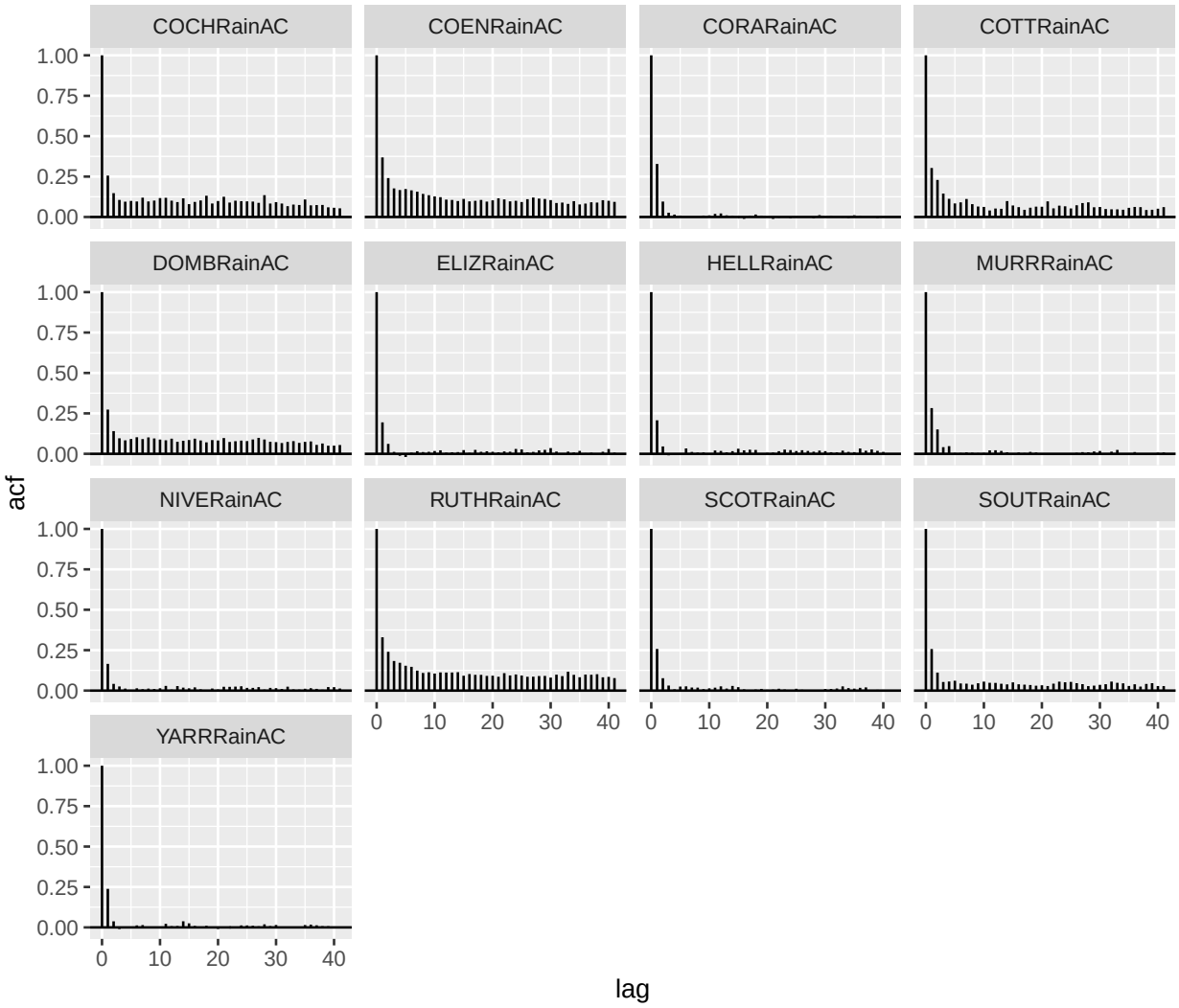

Figure 5: Autocorrelation plot of the station rainfall data

This again shows little difference between the autocorrelation functions, with possibly the gridded rainfall (first graph, Figure 4) being slightly more auto-correlated than the station rainfall data (Second graph, Figure 5).

Storage, combining and publishing data

The last thing to do is to combine the data into a storage data object to be read in during all the analysis. Two different objects need to be developed. One is the daily data, combining the gridded rainfall data and the station rainfall data. The second object combines all the weekly data.

Write data frames out as Rdata and csv

This section creates the on disk csv files and Rdata objects to be used in the rest of the research.

```
gridRain_zoo <- rainAC_zoo

save(GridRainAllDataout,
     file="data/DailyDataIncludingGridded.Rdata")
write.csv(GridRainAllDataout,
          file="data/DailyDataIncludingGridded.csv",
          row.names=F)

# load metadata
zz <- file("Data/README_DataDescription.txt","w+")
text <- c("Metadata Climate Change in Streamflow (MD project)",
"authors: R.Willem Vervoort, Michaela Dolk, Floris van Ogtrop",
"institution: Centre for Carbon Water and Food, Sydney Institute of
Agriculture, The University of Sydney",
"contact: willem.vervoort@sydney.edu.au",
"The data in this project are sourced from the Bureau of Meteorology
website. ",
"They are based on a sample of the hydrological reference stations and
closest rainfall and high quality temperature stations as well as gridded
rainfall data",
"These stations are given in CatchmentCharact.csv and in the Stations R
object in the Rdata file",
"The objects in the Rdata file are:",
"This metadata file as a text object, called README_datadescribe",
"Stations a dataframe with the catchment characteristcs",
"flow_rain_maxT_weekly: a dataframe with column headers Date, decade,
station, Flow, Rain, maxT, stations are stacked",
"flow_zoo: all flow data for the catchments as a zoo data frame, 13
catchments in columns",
"rain_zoo: all rain data for the catchments as a zoo data frame, 13
catchments in columns",
"maxT_zoo: all maxT data for the catchments as a zoo data frame, 13
catchments in columns",
"gridRain_zoo: all gridded rainfall data for the catchments as a zoo data frame,
13 catchments in columns",
"GridRainAllDataout: all daily rainfall including gridded data stacked
catchments",
"The objects in the zip file are:",
"This metadata file",
```

```
"CatchmentCharact.csv",
"flow_rain_maxT_weekly.csv: a dataframe with column headers Date, decade,
station, Flow, Rain, maxT, gridRain stations are stacked",
"DailyDataIncludingGridded.csv: the daily rainfall data included gridded
as a stacked dataframe with 3 columns")
writeLines(text,zz)
README_DataDescribe <- readLines(zz)
close(zz)

# write as an RData file
save("README_DataDescribe","flow_rain_maxT_weekly","Stations","flow_zoo",
      "rain_zoo","maxT_zoo","gridRain_zoo","GridRainAllDataout",
      file="Data/ClimCh_project_MD.Rdata")

write.csv(flow_rain_maxT_weekly,file="data/flow_rain_maxT_weekly.csv",
          row.names=F)

# create also a zip file
zip("Data/ClimCh_project_MD.zip",
    files = c("data/README_DataDescription.txt",
              "data/flow_rain_maxT_weekly.csv",
              "data/DailyDataIncludingGridded.csv",
              "data/CatchmentCharact.csv"))
```