

# Non-parametric Elasticity following Chiew (2006)

*Willem Vervoort, Michaela Dolk & Floris van Ogtrop*

*2017-09-05*

```
# root dir
knitr::opts_knit$set(root.dir = "C:/Users/rver4657/ownCloud/Virtual Experiments/VirtExp")
knitr::opts_chunk$set(echo = TRUE)
# LOAD REQUIRED PACKAGES # #####
library(pander)
library(tidyverse)
library(xts)
library(zoo)
library(ggplot2)
library(reshape2)
```

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 is the fifth part of the series that calculates elasticities following the method described in the paper by Chiew (2006).

FRANCIS H. S. CHIEW (2006) Estimation of rainfall elasticity of streamflow in Australia, Hydrological Sciences Journal, 51:4, 613-625, DOI: 10.1623/hysj.51.4.613

This method involves developing a several climate series which are linearly scaled by different percentages. The climate variables that are scaled are rainfall and potential ET in the original paper, while in this paper we scale rainfall and maximum Temperature (as a proxy for potential ET). The scaled climate series are then used for prediction in the model to derive climate elasticities based on the modelled monthly streamflow. This was done by regressing:

$$\delta Q = \epsilon_P \delta P + \epsilon_{PET} \delta PET$$

In addition, a non-parametric estimator using the annual data was used to also calculate the sensitivity to P. This estimator was:

$$\epsilon_P = \text{median}\left(\frac{Q_t - \bar{Q}}{P_t - \bar{P}} \frac{\bar{P}}{\bar{Q}}\right)$$

So the aim of this document is to calculate the non-parametric epsilon based on the gridded and station rainfall data and the original flow data for the catchments.

## Loading the data

```
load("data/DailyDataIncludingGridded.Rdata")
load("data/ClimCh_project_MD.Rdata")
# correct the column name of maxT in GridRainAllDataout
colnames(GridRainAllDataout)[5] <- "MaxT"
# change to tibble
```

```

GridAlldata <- as.tibble(cbind(GridRainAllDataout,
                              Date=rep(time(flow_zoo),nrow(Stations))))
# now use spread
GridAlldata_wide <- spread(GridAlldata[,c("Station","gridRain","Date")],
                           key=Station, value=gridRain)
Gridrain_zoo <- zoo(GridAlldata_wide[,2:14],order.by=time(flow_zoo))

```

## Aggregate flow and rain to annual data

First aggregate all the station data, as this includes the flow data. Then aggregate the gridded rainfall data.

```

# station data for flow
annualF <- aggregate(flow_zoo,format(time(flow_zoo),"%Y"),sum,na.rm=T)
mean_annualF <- apply(annualF,2,mean,na.rm=T)
# rainfall
annualR <- aggregate(rain_zoo,format(time(rain_zoo),"%Y"),sum,na.rm=T)
mean_annualR <- apply(annualR,2,mean,na.rm=T)

# Maximum temperature
annualT <- aggregate(maxT_zoo,format(time(rain_zoo),"%Y"),mean,na.rm=T)
mean_annualT <- apply(annualT,2,mean,na.rm=T)

# grid rainfall
annualGridR <- aggregate(Gridrain_zoo,format(time(Gridrain_zoo),"%Y", na.rm=T),
                        sum)
mean_annualGridR <- apply(annualGridR,2,mean, na.rm=T)

```

## Check anomalies for trends in rainfall

We need to check if there is no consistent trend in the rainfall anomalies that we need to consider. If there is a significant trend then this might also be an indication.

```

# Check cumulative running anomalies to make missing values are not a big thing
mean_dailyR <- apply(rain_zoo,2,mean,na.rm=T)
mean_dailyGridR <- apply(Gridrain_zoo,2,mean,na.rm=T)
store <- list()
storeGrid <- list()

# cumulative sums
for (i in 1:ncol(rain_zoo)) {
  x <- rain_zoo[,i]
  x1 <- Gridrain_zoo[,i]
  miss <- is.na(rain_zoo[,i])
  miss_g <- is.na(Gridrain_zoo[,i])
  #x[miss] <- 0
  # Station rainfall
  cs <- cumsum(ifelse(is.na(rain_zoo[,i]-mean_dailyR[i])==T,
                    0, x - mean_dailyR[i]))
  cs[miss] <- NA
  store[[i]] <- data.frame(Date=time(rain_zoo),data=as.numeric(cs),
                          station=rep(Stations[i,1],length(cs)))
  # Gridded rainfall

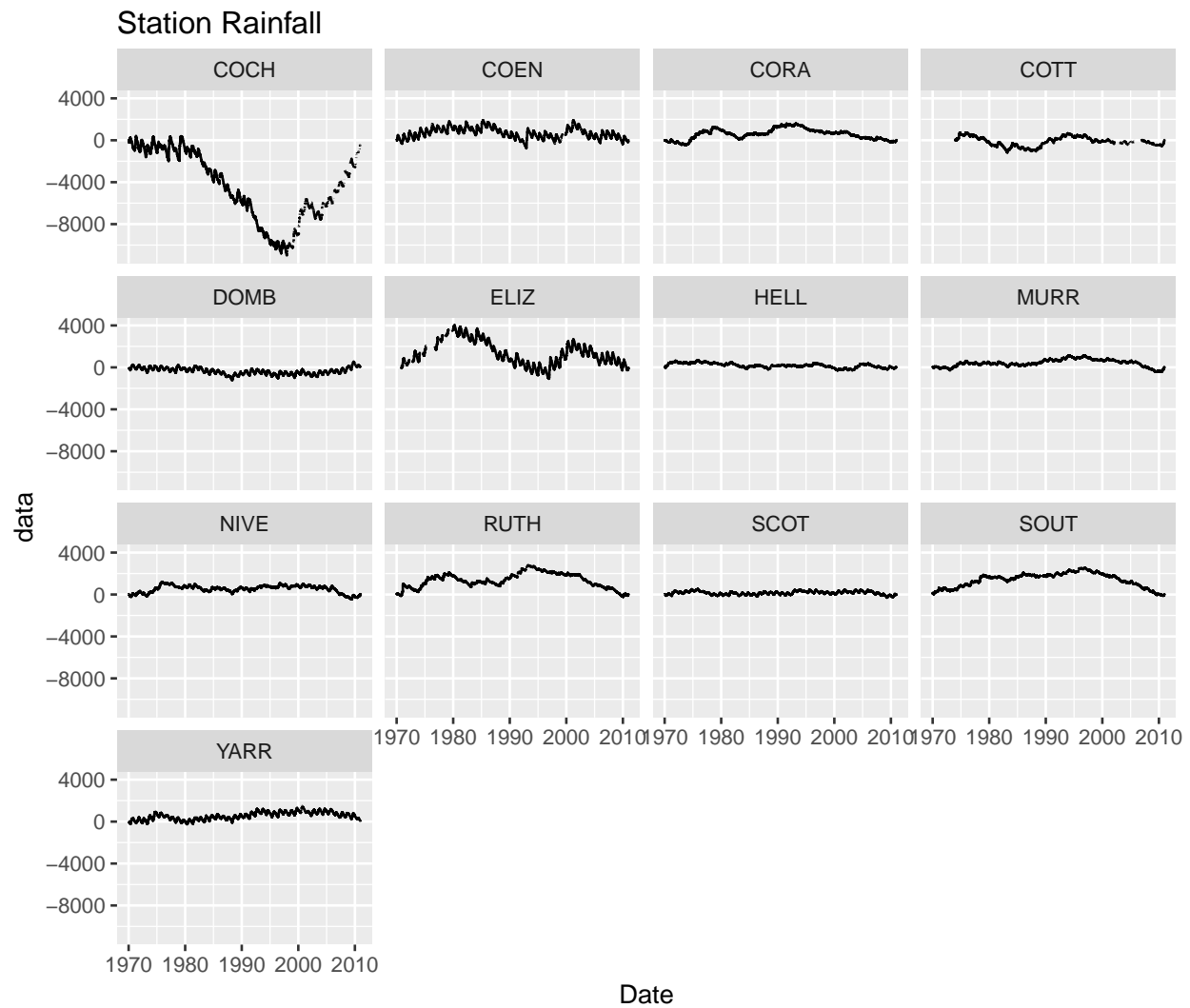
```

```

cs <- cumsum(ifelse(is.na(Gridrain_zoo[,i])-mean_dailyGridR[i])==T,
  0, x1 - mean_dailyGridR[i]))
cs[miss_g] <- NA
storeGrid[[i]] <- data.frame(Date=time(Gridrain_zoo),data=as.numeric(cs),
  station=rep(Stations[i,1],length(cs)))
}

# Make a simple plot
plot.df <- data.frame(do.call(rbind,store))
plot_g.df <- data.frame(do.call(rbind,storeGrid))
ggplot(plot.df,aes(x=Date,y=data)) + geom_line() + facet_wrap(~station) +
  ggtitle("Station Rainfall")

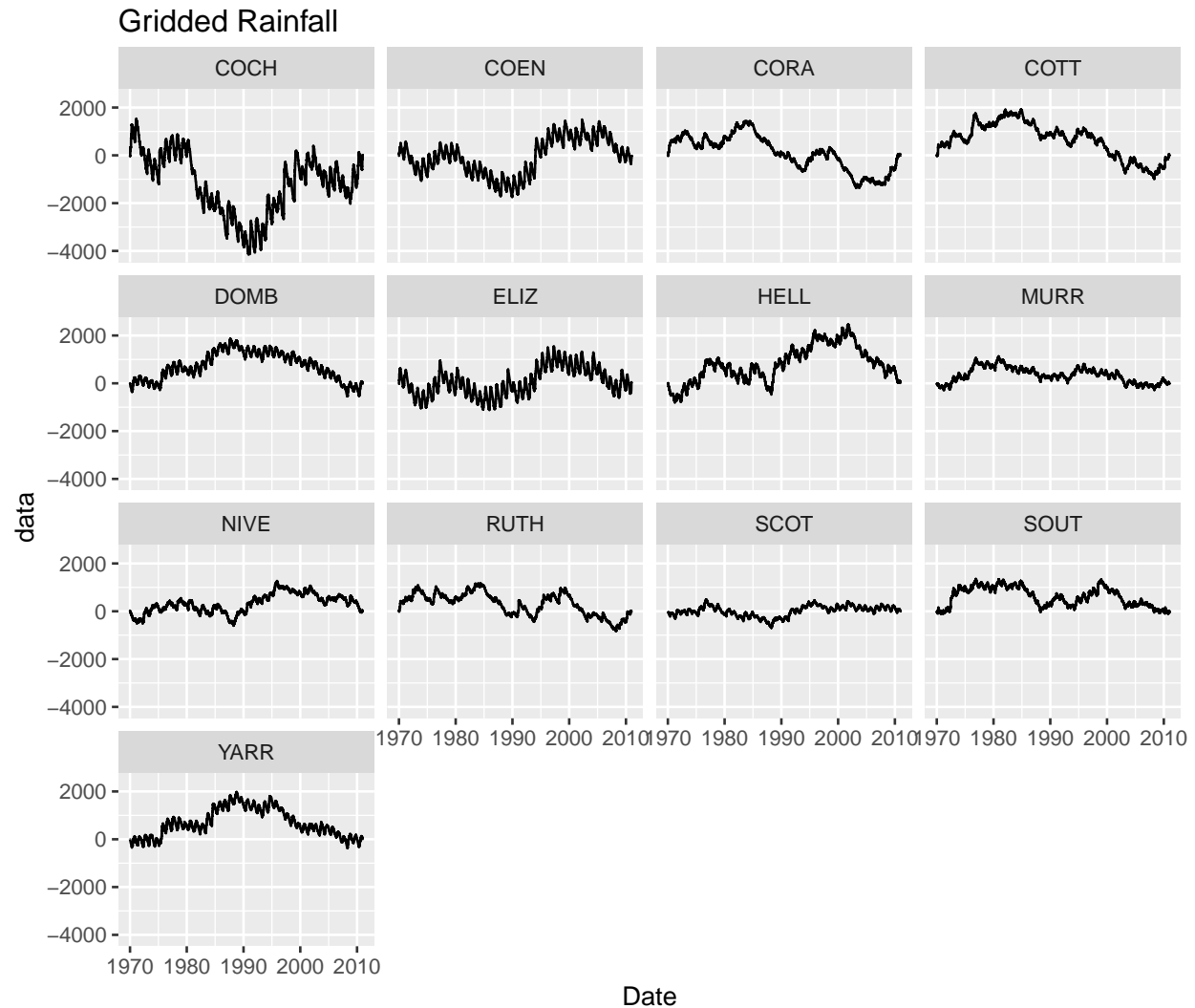
```



```

ggplot(plot_g.df,aes(x=Date,y=data)) + geom_line() + facet_wrap(~station) +
  ggtitle("Gridded Rainfall")

```



*# No clear indication of a trend in any of the stations*

The anomaly plots, while showing significant variation across the decades (particularly for the COCH station data), show no clear trends up or down, but more a difference between decades, wetting up in some decades and drying out in other decades.

## Calculating $e_p$

Non parametric calculation of the elasticity  $e_p$

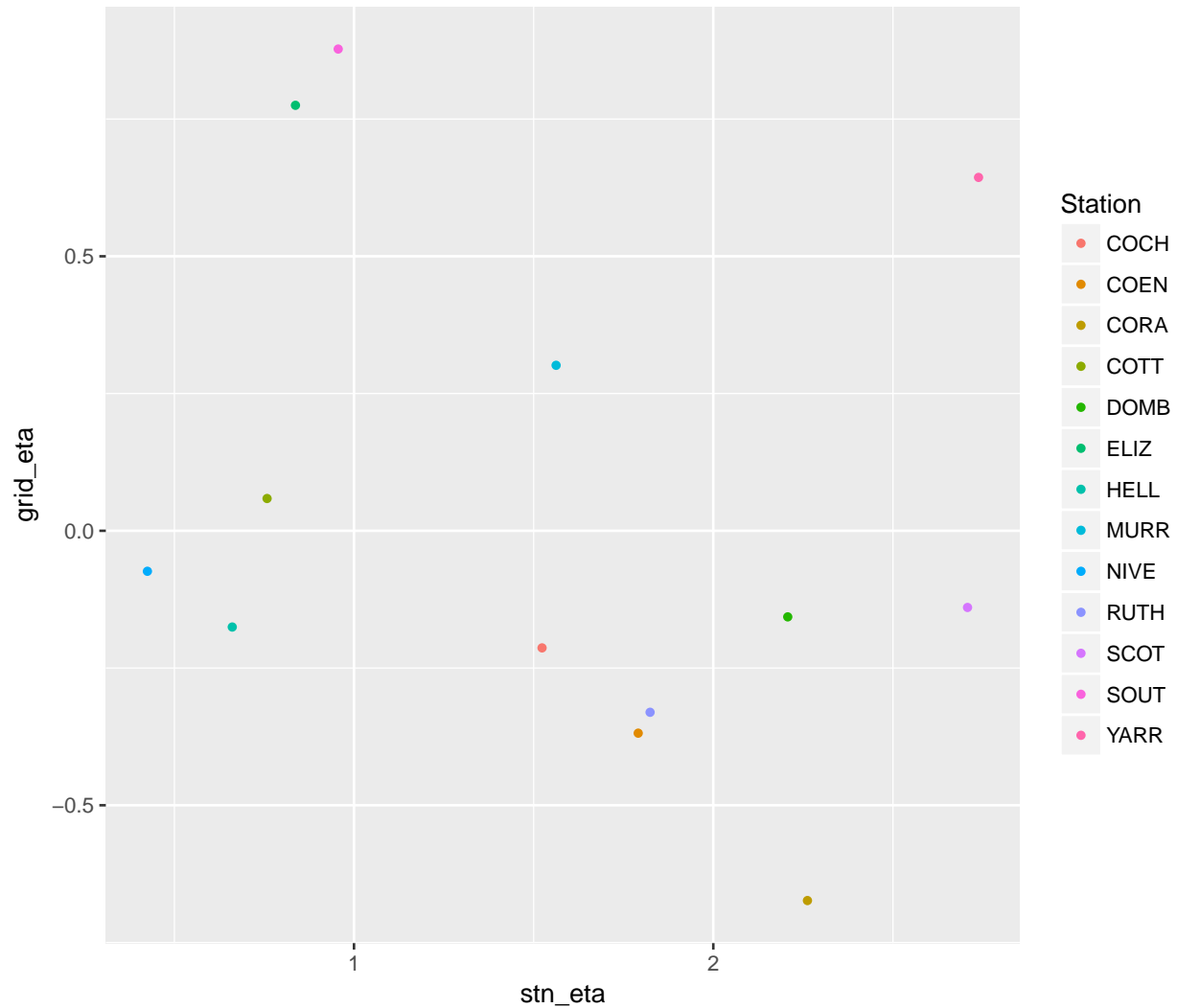
```
eta_p <- function(Q,P,meanQ,meanP) {median(((Q-meanQ)/(P-meanP)*meanP/meanQ))}

# station rainfall
out <- list()
for (i in 1:ncol(annualF)) {
  out[[i]] <- eta_p(annualF[,i], annualR[,i],mean_annualF[i],mean_annualR[i])
}
non_par_eta <- data.frame(stn = Stations[,1],eta_p=do.call(c,out))
```

```
# gridded rainfall
out2 <- list()
for (i in 1:ncol(annualF)) {
  out2[[i]] <- eta_p(annualF[,i],
                    annualGridR[,i],mean_annualF[i],mean_annualGridR[i])
}
non_par_eta_g <- data.frame(stn = Stations[,1],eta_p=do.call(c,out2))
```

Plot the sensitivity based on the station rainfall against the sensitivity based on the gridded rainfall.

```
# a simple plot
plot.df <- data.frame(Station=Stations[,1], stn_eta = non_par_eta$eta_p,
                    grid_eta = non_par_eta_g$eta_p)
p <- ggplot(plot.df,aes(x=stn_eta, y=grid_eta)) +
  geom_point(aes(colour=Station), pch=16)
print(p)
```



There clearly is a large difference between the non-parametric epsilon calculated from the gridded data, compared to epsilon calculated from the station data. The station data is much more in alignment with the original paper (Chiew, 2006), while the gridded data is clearly much smoother and therefore does not indicate

the amplifications that are visible in the station data.

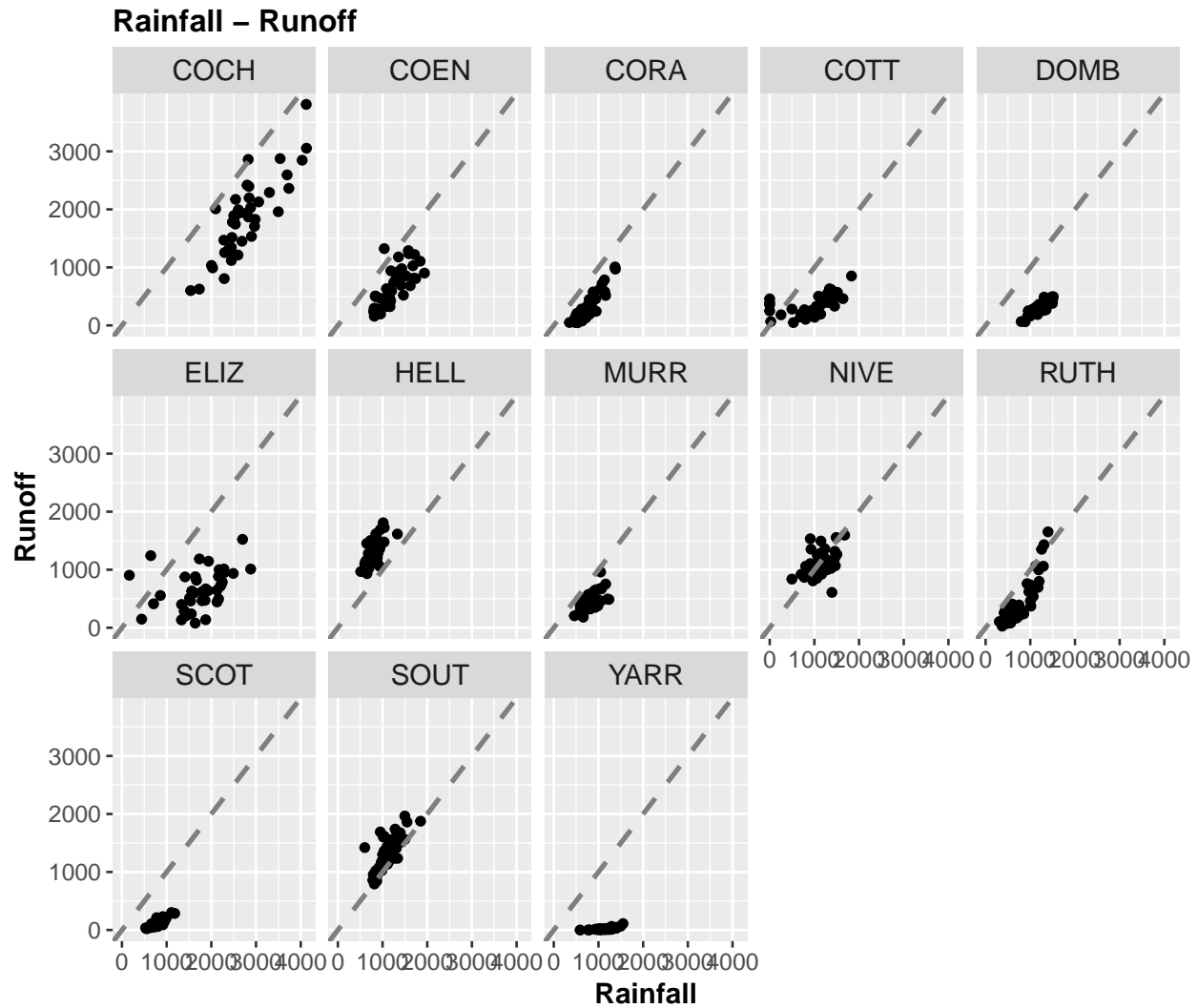
## Rainfall runoff plots for annual data (station only)

For a sanity check we plot the rainfall against the runoff data to understand the runoff coefficients.

```
# First rearrange the annual data to a dataframe
flow_an_stack <- stack(as.data.frame(annualF))
rain_an_stack <- stack(as.data.frame(annualR))
maxT_an_stack <- stack(as.data.frame(annualT))

RR_df <- data.frame(stn =substr(flow_an_stack$ind,1,4),flow=flow_an_stack$values,
                    rain=rain_an_stack$values,maxT=maxT_an_stack$values,
                    trend = rep(1:nrow(annualF),ncol(annualF)))

p <- ggplot(RR_df, aes(x = rain, y = flow)) +
  geom_point(size=1.5,col="black") +
  geom_abline(intercept=0, slope=1,col="grey50",lty=2,lwd=1) +
  facet_wrap(~ stn,ncol=5) +
  guides(col = guide_legend(nrow = 3))
p <- p + ggtitle("Rainfall - Runoff") +
  theme(plot.title = element_text(lineheight=.8, face="bold"))+
  xlab("Rainfall") +
  theme(axis.title.x = element_text(face="bold", size=12),
        axis.text.x = element_text(size=10)) +
  ylab("Runoff") +
  theme(axis.title.y = element_text(face="bold", size=12),
        axis.text.y = element_text(size=10)) +
  theme(legend.text = element_text( size = 10))+
  theme(legend.title = element_text(size=12, face="bold")) +
  theme(strip.text.x = element_text(size=12))
print(p)
```



```
# publication quality
tiff("../Manuscript/Figure2_RainfallRunoffPlot.tif",
      width=16*480,height=12*480, res=600, compression="lzw")
print(p)
dev.off()
```

```
## pdf
## 2
```

Saving output for later

```
# Save the data
write.csv(RR_df,
          file="Data/RainfallRunoff.csv",row.names=F)
# save output for document 7
write.csv(plot.df,"Data/non_par_eta.csv", row.names=F)
```