Modelled vs Field Data

# Setting up:

Load libraries:

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(ggplot2)  
library(ncdf4)  
library(ncdf4.helpers)

## Warning: package 'ncdf4.helpers' was built under R version 3.4.4

library(reshape2)

Set working directory:

path <- "D:/Maria/Proyectos/ECOPOTENTIAL\_FUTURE\_PROJECTIONS/" # María  
# path <- "..." # Ricardo

# Description

Once models are compared with each other to assess spread and uncertainty (see "Model intercomparison.Rmd"), modelled data should be compared with field observations to assess model performance/model error. Field observations data will be downloaded from Linaria database. Documentation of this process (selection of stations, variables, queries, filters,...) can be found in "Field\_Data.Rmd".

Once field observations are downloaded, modelled and field data can be compared. Although they are not downloaded yet, I start preparing the code for the comparison in further sections with a test dataset. Since several variables have been calculated (yearmean, season mean, month mean, day min, day max), several sections will be needed. At the moment, since the test dataset refers to monthly means, analysis is done for this case only.

For notes on how such comparison should be done see "Chapter 16. Assessing Model Performance" (Ecological Forecasting, M. Dietze). At minimum, calculate RMSE, correlation, bias and ratio between standard deviations.

# Extract pixels indexes where there are stations

We select only the stations where we know some variables are being measured. As "Field\_Data.Rmd" advances, *this list could be modified, be careful*

Stations:

my\_stations <- "filtered\_stations.csv"  
my\_stations <- read.csv(paste(path, "Field\_Observations/", my\_stations, sep = ""), sep = ",")

Modelled data - All models have the same long and lat, I open a random one:

CNRM <- "temp\_CNRM\_historical\_monmean.nc"  
cnrm\_nc <- nc\_open(paste(path, "Outputs\_201807/", CNRM, sep = ""))

Extract pixel indexes:

lon <- ncvar\_get(cnrm\_nc, "lon")  
lat <- ncvar\_get(cnrm\_nc, "lat", verbose = F)  
  
# nc files have associated indexes for latitude and longitud to identy each pixel, so each station will have associated the indexes of the pixel where it stands  
my\_stations$pixel\_lat\_index <- c()  
my\_stations$pixel\_lon\_index <- c()  
  
for (i in 1:length(my\_stations$cli\_estacion\_id)) {  
 lat\_value <- my\_stations$lat[i]  
 lon\_value <- my\_stations$lon[i]  
   
 my\_stations$pixel\_index\_lat[i] <- which.min(abs(lat - lat\_value))  
 my\_stations$pixel\_index\_lon[i] <- which.min(abs(lon - lon\_value))  
}

# Generate data frame containing time series of all stations

Test dataset of field observations are **monthly** means, so I do the analysis for the monthly means, but same analysis should be performed for all other metrics.

*Note* should this df has the mean of the 5 models or the date of the 5 models separately? Options:

* df containing the 5 models (then an extra column indicating the model would be necessary, or each model data would be in one column),
* df containing the average of the 5 models (then such average should be calculated, and maybe an extra parameter indicating variability should be included)

I decide to construct the df containing one column per model + mean + sd among values, so that when I include the observations as an extra column comparison can be easily done for each model individually or for the mean of them.

Load modelled data:

CNRM <- "temp\_CNRM\_historical\_monmean.nc"  
cnrm\_nc <- nc\_open(paste(path, "Outputs\_201807/", CNRM, sep = ""))  
  
ICHEC <- "temp\_ICHEC\_historical\_monmean.nc"  
ichec\_nc <- nc\_open(paste(path, "Outputs\_201807/", ICHEC, sep = ""))  
  
IPSL <- "temp\_IPSL\_historical\_monmean.nc"  
ipsl\_nc <- nc\_open(paste(path, "Outputs\_201807/", IPSL, sep = ""))  
  
MOHC <- "temp\_MOHC\_historical\_monmean.nc"  
mohc\_nc <- nc\_open(paste(path, "Outputs\_201807/", MOHC, sep = ""))  
  
MPI <- "temp\_MPI\_historical\_monmean.nc"  
mpi\_nc <- nc\_open(paste(path, "Outputs\_201807/", MPI, sep = ""))

Generate models time series

timeseries <- data.frame()  
  
for (i in 1:length(my\_stations$cli\_estacion\_id)) {  
 # Collect "tas" time series for each model  
 tas\_cnrm <- nc.get.var.subset.by.axes(cnrm\_nc, "tas", axis.indices = list(X = my\_stations$pixel\_index\_lon[i], Y = my\_stations$pixel\_index\_lat[i]))  
 tas\_ichec <- nc.get.var.subset.by.axes(ichec\_nc, "tas", axis.indices = list(X = my\_stations$pixel\_index\_lon[i], Y = my\_stations$pixel\_index\_lat[i]))  
 tas\_ipsl <- nc.get.var.subset.by.axes(ipsl\_nc, "tas", axis.indices = list(X = my\_stations$pixel\_index\_lon[i], Y = my\_stations$pixel\_index\_lat[i]))  
 tas\_mohc <- nc.get.var.subset.by.axes(mohc\_nc, "tas", axis.indices = list(X = my\_stations$pixel\_index\_lon[i], Y = my\_stations$pixel\_index\_lat[i]))  
 tas\_mpi <- nc.get.var.subset.by.axes(mpi\_nc, "tas", axis.indices = list(X = my\_stations$pixel\_index\_lon[i], Y = my\_stations$pixel\_index\_lat[i]))  
   
 # Collect "time" time series for each model - although all are monthly means, apparently the time point to which the data is linked is not the same for all models. However, they do refer to the same month and year, so only one time timeseries will be used  
 tas\_cnrm\_time <- nc.get.time.series(cnrm\_nc, v = "tas",  
 time.dim.name = "time")  
 # tas\_ichec\_time <- nc.get.time.series(ichec\_nc, v = "tas",  
 # time.dim.name = "time")  
 # tas\_ipsl\_time <- nc.get.time.series(ipsl\_nc, v = "tas",  
 # time.dim.name = "time")  
 # tas\_mohc\_time <- nc.get.time.series(mohc\_nc, v = "tas",  
 # time.dim.name = "time")  
 # tas\_mpi\_time <- nc.get.time.series(mpi\_nc, v = "tas",  
 # time.dim.name = "time")  
   
 temp <- data\_frame(time\_cnrm = tas\_cnrm\_time,  
 tas\_cnrm = as.vector(tas\_cnrm),  
 # time\_ichec = tas\_ichec\_time,  
 tas\_ichec = as.vector(tas\_ichec),  
 # time\_ipsl = tas\_ipsl\_time,  
 tas\_ipsl = as.vector(tas\_ipsl),  
 # time\_mohc = tas\_mohc\_time,  
 tas\_mohc = as.vector(tas\_mohc),  
 # time\_mpi = tas\_mpi\_time,  
 tas\_mpi = as.vector(tas\_mpi),  
 codigo = my\_stations$codigo[i])  
   
 timeseries <- rbind(timeseries, temp)  
}

Fetch time series

timeseries <- timeseries %>%  
 mutate(time\_cnrm = as.Date(format(time\_cnrm, "%Y-%m-%d"))) %>%  
 mutate(year = as.integer(format(time\_cnrm,"%Y"))) %>%  
 mutate(month = as.integer(format(time\_cnrm,"%m"))) %>%  
 select(codigo, time\_cnrm, year, month, tas\_cnrm, tas\_ichec, tas\_ipsl, tas\_mohc, tas\_mpi) %>%  
 mutate(codigo = as.character(codigo))

Add field observations to time series dataframe

clean\_mfo <- read.csv(paste(path, "Field\_Observations/clean\_mfo.csv", sep = ""))  
  
clean\_mfo <- clean\_mfo %>%  
 mutate(obs\_value = value) %>%  
 select(codigo, year, month, obs\_value) %>%  
 mutate(codigo = as.character(codigo))  
  
full\_ts <- inner\_join(clean\_mfo, timeseries)

## Joining, by = c("codigo", "year", "month")

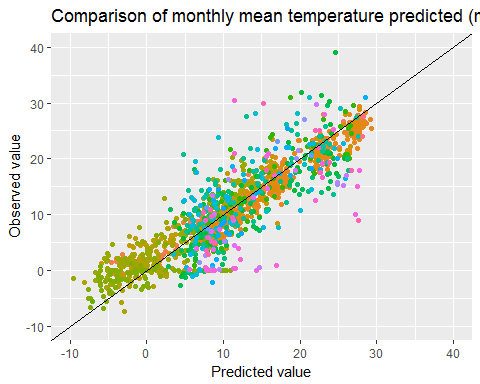
Add metrics

full\_ts <- full\_ts %>%  
 mutate(modelled\_mean = ((tas\_cnrm + tas\_ichec + tas\_ipsl + tas\_mohc + tas\_mpi) / 5)) %>%  
 mutate(modelled\_sd = (sqrt((((tas\_cnrm - modelled\_mean) ^ 2) + ((tas\_ichec - modelled\_mean) ^ 2) + ((tas\_ipsl - modelled\_mean) ^ 2) + ((tas\_mohc - modelled\_mean) ^ 2) + ((tas\_mpi - modelled\_mean) ^ 2)) / 4)))

# Plotting - qualitative exploration

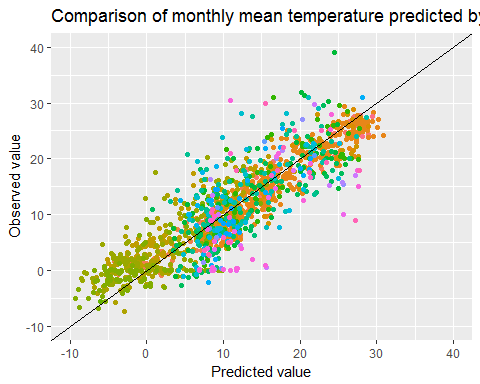
Modelled vs observations: mean of models

ggplot(full\_ts, aes(x = modelled\_mean, y = obs\_value, colour = codigo)) +  
 geom\_point() +   
 xlab("Predicted value") + ylab("Observed value") +   
 theme(legend.position="none") +  
 ggtitle("Comparison of monthly mean temperature predicted (mean across models) and observed values") +  
 scale\_y\_continuous(limits=c(-10, 40)) +  
 scale\_x\_continuous(limits=c(-10, 40)) +  
 geom\_abline(slope = 1, intercept = 0)



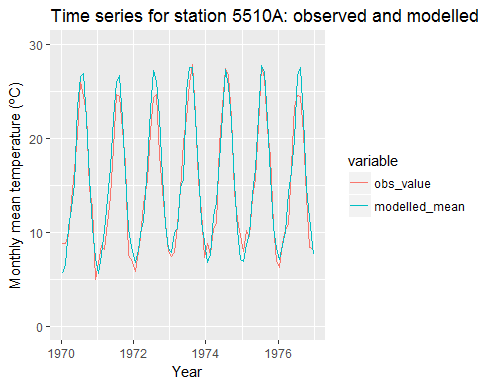
Modelled vs observations: one model

ggplot(full\_ts, aes(x = tas\_cnrm, y = obs\_value, colour = codigo)) +  
 geom\_point() +   
 xlab("Predicted value") + ylab("Observed value") +   
 theme(legend.position="none") +  
 ggtitle("Comparison of monthly mean temperature predicted by model CNRM and observed") +  
 scale\_y\_continuous(limits=c(-10, 40)) +  
 scale\_x\_continuous(limits=c(-10, 40)) +  
 geom\_abline(slope = 1, intercept = 0)



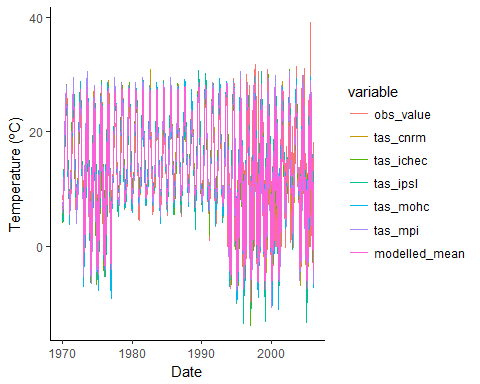
Modelled vs observations: one-station time series

selection <- full\_ts %>%  
 filter(codigo == "5510A") %>%  
 select(time\_cnrm, obs\_value, modelled\_mean) %>%  
 melt(id = c("time\_cnrm"))  
  
ggplot(data = selection, aes(x = time\_cnrm, y = value, colour = variable)) +  
 geom\_line() +   
 xlab("Year") + ylab("Monthly mean temperature (ºC)") +   
 theme() +  
 ggtitle("Time series for station 5510A: observed and modelled") +  
 scale\_y\_continuous(limits=c(0, 30))



Modelled vs observations: all-stations time series all models and mean

reshaped\_ts <- melt(full\_ts, id = c("codigo", "year", "month", "time\_cnrm")) %>%  
 filter(variable != "modelled\_sd")  
   
ggplot(reshaped\_ts, aes(x = time\_cnrm, y = value, colour = variable)) +  
 geom\_line() +  
 xlab("Date") + ylab("Temperature (ºC)") +  
 theme\_classic()

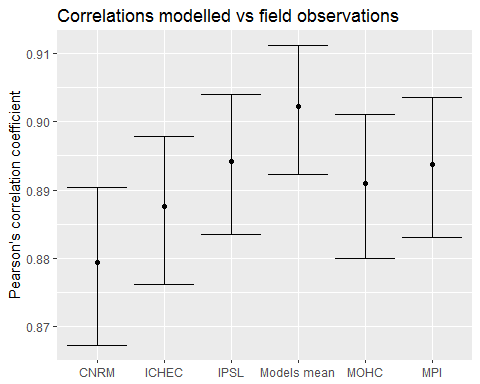


## Quantitative analysis

Following "Chapter 16. Assessing Model Performance" (Ecological Forecasting, M. Dietze).

Correlations: per model and against modelled mean

a <- cor.test(full\_ts$obs\_value, full\_ts$tas\_cnrm)  
b <- cor.test(full\_ts$obs\_value, full\_ts$tas\_ichec)  
c <- cor.test(full\_ts$obs\_value, full\_ts$tas\_ipsl)  
d <- cor.test(full\_ts$obs\_value, full\_ts$tas\_mohc)  
e <- cor.test(full\_ts$obs\_value, full\_ts$tas\_mpi)  
f <- cor.test(full\_ts$obs\_value, full\_ts$modelled\_mean)   
  
names <- c("CNRM", "ICHEC", "IPSL", "MOHC", "MPI", "Models mean")  
correlations <- c(a$estimate, b$estimate, c$estimate, d$estimate, e$estimate, f$estimate)  
ci\_1 <- c(a$conf.int[1], b$conf.int[1], c$conf.int[1], d$conf.int[1], e$conf.int[1], f$conf.int[1])  
ci\_2 <- c(a$conf.int[2], b$conf.int[2], c$conf.int[2], d$conf.int[2], e$conf.int[2], f$conf.int[2])  
  
cor\_table <- data.frame(names, correlations, ci\_1, ci\_2)  
  
ggplot(data = cor\_table) +  
 geom\_errorbar(mapping=aes(x = names, ymin = ci\_1, ymax = ci\_2)) +  
 geom\_point(mapping=aes(x = names, y = correlations)) +  
 xlab(NULL) +  
 ylab("Pearson's correlation coefficient") +  
 ggtitle("Correlations modelled vs field observations")



Root Mean Square Error (RMSE) - calculated only for the modelled mean

meanofdata <- mean(full\_ts$obs\_value)  
  
rmse <- full\_ts %>%  
 select(codigo, year, month, obs\_value, modelled\_mean, modelled\_sd) %>%  
 mutate(sq\_error = (modelled\_mean - obs\_value) ^ 2) %>%  
 mutate(deviationfromdata = (obs\_value - meanofdata) ^ 2)  
  
R\_square <- 1 - (sum(rmse$sq\_error) / sum(rmse$deviationfromdata))  
  
RMSE <- sqrt(sum(rmse$sq\_error)/length(rmse$sq\_error)) # is this correct?  
  
bias <- mean(rmse$modelled\_mean) - mean(rmse$obs\_value)  
  
sd\_ratio <- sd(rmse$modelled\_mean) / sd(rmse$obs\_value) # the models have more variability than what it is observed, they predict more fluctuations than the ones which actually occur.

# Notes: how to work as nc files

Open .nc files

# cnrm\_nc <- nc\_open(paste(path, CNRM, sep = ""))  
# ichec\_nc <- nc\_open(paste(path, ICHEC, sep = ""))  
# ipsl\_nc <- nc\_open(paste(path, IPSL, sep = ""))  
# mohc\_nc <- nc\_open(paste(path, MOHC, sep = ""))  
# mpi\_nc <- nc\_open(paste(path, MPI, sep = ""))

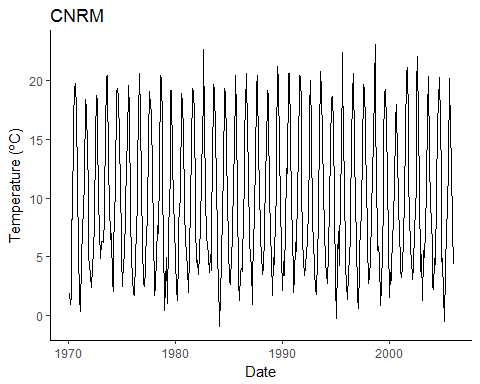
Check variables and nc file structure

# cnrm\_nc  
# attributes(cnrm\_nc)  
# attributes(cnrm\_nc$var)  
#   
# ncvar\_get(cnrm\_nc, "lat")  
# ncvar\_get(cnrm\_nc, "lon")  
# ncvar\_get(cnrm\_nc)  
#   
# max(ncvar\_get(cnrm\_nc, "lon"))  
# min(ncvar\_get(cnrm\_nc, "lon"))  
# max(ncvar\_get(cnrm\_nc, "lat"))  
# min(ncvar\_get(cnrm\_nc, "lat"))

# Create an array from the nc  
cnrm.array <- ncvar\_get(cnrm\_nc, "tas")  
mpi.array <- ncvar\_get(mpi\_nc, "tas")  
  
# Take one slice from the array  
cnrm.slice <- cnrm.array[, , 1]   
mpi.slice <- mpi.array[, , 1]

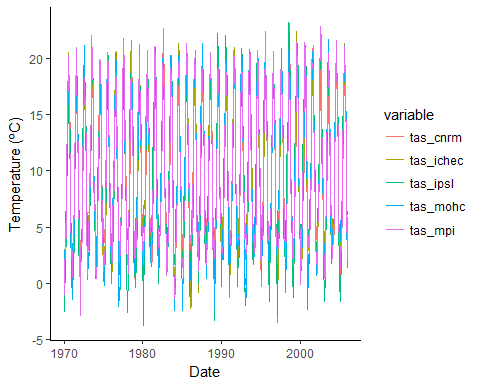
Single model point timeseries plots:

lon <- ncvar\_get(cnrm\_nc, "lon")  
lat <- ncvar\_get(cnrm\_nc, "lat", verbose = F)  
lon\_index <- which.min(abs(lon - -3.500000))   
lat\_index <- which.min(abs(lat - 37.05000))   
  
# cnrm  
tas\_cnrm <- nc.get.var.subset.by.axes(cnrm\_nc, "tas", axis.indices = list(X = lon\_index, Y = lat\_index))  
  
tas\_cnrm\_time <- nc.get.time.series(cnrm\_nc, v = "tas",  
 time.dim.name = "time")  
data\_frame(time = tas\_cnrm\_time,   
 tas = as.vector(tas\_cnrm)) %>%  
 mutate(time = as.Date(format(time, "%Y-%m-%d"))) %>%  
 ggplot(aes(x = time, y = tas)) +   
 geom\_line() +   
 xlab("Date") + ylab("Temperature (ºC)") +   
 theme\_classic() +  
 ggtitle("CNRM")



All models point timeseries dataframe:

tas\_cnrm <- nc.get.var.subset.by.axes(cnrm\_nc, "tas", axis.indices = list(X = lon\_index, Y = lat\_index))  
tas\_ichec <- nc.get.var.subset.by.axes(ichec\_nc, "tas", axis.indices = list(X = lon\_index, Y = lat\_index))  
tas\_ipsl <- nc.get.var.subset.by.axes(ipsl\_nc, "tas", axis.indices = list(X = lon\_index, Y = lat\_index))  
tas\_mohc <- nc.get.var.subset.by.axes(mohc\_nc, "tas", axis.indices = list(X = lon\_index, Y = lat\_index))  
tas\_mpi <- nc.get.var.subset.by.axes(mpi\_nc, "tas", axis.indices = list(X = lon\_index, Y = lat\_index))  
  
ts <- data\_frame(cnrm\_time = tas\_cnrm\_time, tas\_cnrm = as.vector(tas\_cnrm), tas\_ichec = as.vector(tas\_ichec), tas\_ipsl = as.vector(tas\_ipsl), tas\_mohc = as.vector(tas\_mohc), tas\_mpi = as.vector(tas\_mpi)) %>%  
 mutate(cnrm\_time = as.Date(format(cnrm\_time, "%Y-%m-%d")))  
  
melted\_ts <- melt(ts, id = c("cnrm\_time"))  
  
ggplot(melted\_ts, aes(x = cnrm\_time, y = value, colour = variable)) +  
 geom\_line() +  
 xlab("Date") + ylab("Temperature (ºC)") +  
 theme\_classic()



Close all

nc\_close(cnrm\_nc)  
nc\_close(ichec\_nc)  
nc\_close(ipsl\_nc)  
nc\_close(mohc\_nc)  
nc\_close(mpi\_nc)