# SUPPLEMENTARY INFORMATION S3

**Risk and Resilience in the Late Glacial: a Case Study from the Western Mediterranean**

*C Michael Barton, J Emili Aura Tortosa, Oreto Garcia-Puchol, Julien G Riel-Salvatore, Nicolas Gauthier, Margarita Vadillo Conesa, & Geneviève Pothier-Bouchard*

*Last Updated: 2017-07-26*

## Contents



## Introduction

This R Markdown script includes the workflow for all paleoclimate model analyses used in the paper *Risk and Resilience in the Late Glacial: A Case Study from the Western Mediterranean* (Michael Barton corresponding author), published in Quaternary Science Reviews.

In this two-part analysis we first compare temporal patterns of temperature and precipitation, derived from a transient paleoclimate simulation, at three points in the west Mediterranean. Then we analyze the spatial pattern of climate change from the LGM to the Mid Holocene using an ensemble of downscaled equilibrium time-slice paleoclimate simulations.

## Data sources

This R Markdown script requires raster data of observed present-day temperature and precipitation, large scale atmospheric reanalysis data for the present day, TraCE-21k simulation outputs, and an ensemble of monthly atmospheric climatologies from the PMIP3 modeling project. More information on these datasets can be found in the references below, and each can be downloaded by following the links above. The code in the following sections begins with these raw data files and details all preprocessing and analysis steps, but in the interest of reproducibility we include .RDS files of the preprocessed data so that the analysis can proceed if the raw data are unavailable.

**Present-day high resolution observed data (MODIS and CHIRPS derived products)** Deblauwe V., Droissart V., Bose R., Sonké B., Blach-Overgaard A., Svenning J-C, Wieringa J. J., Ramesh B. R., Stévart T. & Couvreur T. L. P. 2016. *Remotely sensed temperature and precipitation data improve species distribution modeling in the tropics*. Global Ecology and Biogeography. 25(4): 443-454

**Present-day coarse resolution reanalysis data (ERA-Interim reanalysis data, monthly means of daily means and daily forecast accumulations, 1979-2010)** Dee, D. P., Uppala, S. M., Simmons, A. J., Berrisford, P., Poli, P., Kobayashi, S., Andrae, U., Balmaseda, M. A., Balsamo, G., Bauer, P., Bechtold, P., Beljaars, A. C. M., van de Berg, L., Bidlot, J., Bormann, N., Delsol, C., Dragani, R., Fuentes, M., Geer, A. J., Haimberger, L., Healy, S. B., Hersbach, H., Hólm, E. V., Isaksen, L., Kållberg, P., Köhler, M., Matricardi, M., McNally, A. P., Monge-Sanz, B. M., Morcrette, J.-J., Park, B.-K., Peubey, C., de Rosnay, P., Tavolato, C., Thépaut, J.-N. and Vitart, F. 2011. *The ERA-Interim reanalysis: configuration and performance of the data assimilation system*. Q.J.R. Meteorol. Soc., 137: 553–597.

**Transient paleoclimate model output (TraCE21k decadal averages)** He, Feng. *Simulating Transient Climate Evolution of the Last deglaciation with CCSM3*. PhD diss., University of Wisconsin-Madison, 2010.

**Time-slice paleoclimate model outputs (PMIP3 monthly climatologies)** Search query: Project: PMIP3 | Model: BCC-CSM1.1, CCSM4, CNRM-CM5, GISS-E2-R, IPSL-CM5A-LR, MIROC-ESM, MRI-CGCM3 | Time frequency: monClim | Experiment: LGM and midHolocene | Realm: atmos

Braconnot P, Harrison S P, Kageyama M, Bartlein P J, Masson-Delmotte V, Abe-Ouchi A, Otto-Bliesner B and Zhao Y. 2012. *Evaluation of climate models using palaeoclimatic data*. Nat. Clim. Change 2: 417–24

## Setup

Load all the packages needed for this analysis.

```
library(ncdf4) # import GCM data
library(rgdal) # read GCM data
library(raster) # process GCM data
library(rasterVis) # plotting GCM data
library(tidyverse) # data management and plotting
library(magrittr) # pipes for code readability
library(EMD) # calculate trends in the data
```

```
library(dismo) # for latitudinally weighted samples
library(mgcv) # fit GAM for downscaling
```

## Temporal Patterns: TraCE-21k

### Sample Locations

Create a matrix with the coordinates for the three locations of interest in the west Mediterranean.

```
samp.pts <- matrix(c(0, 40, 4, 44, 12, 46, 14, 43),
                   ncol = 2, byrow = T)
```

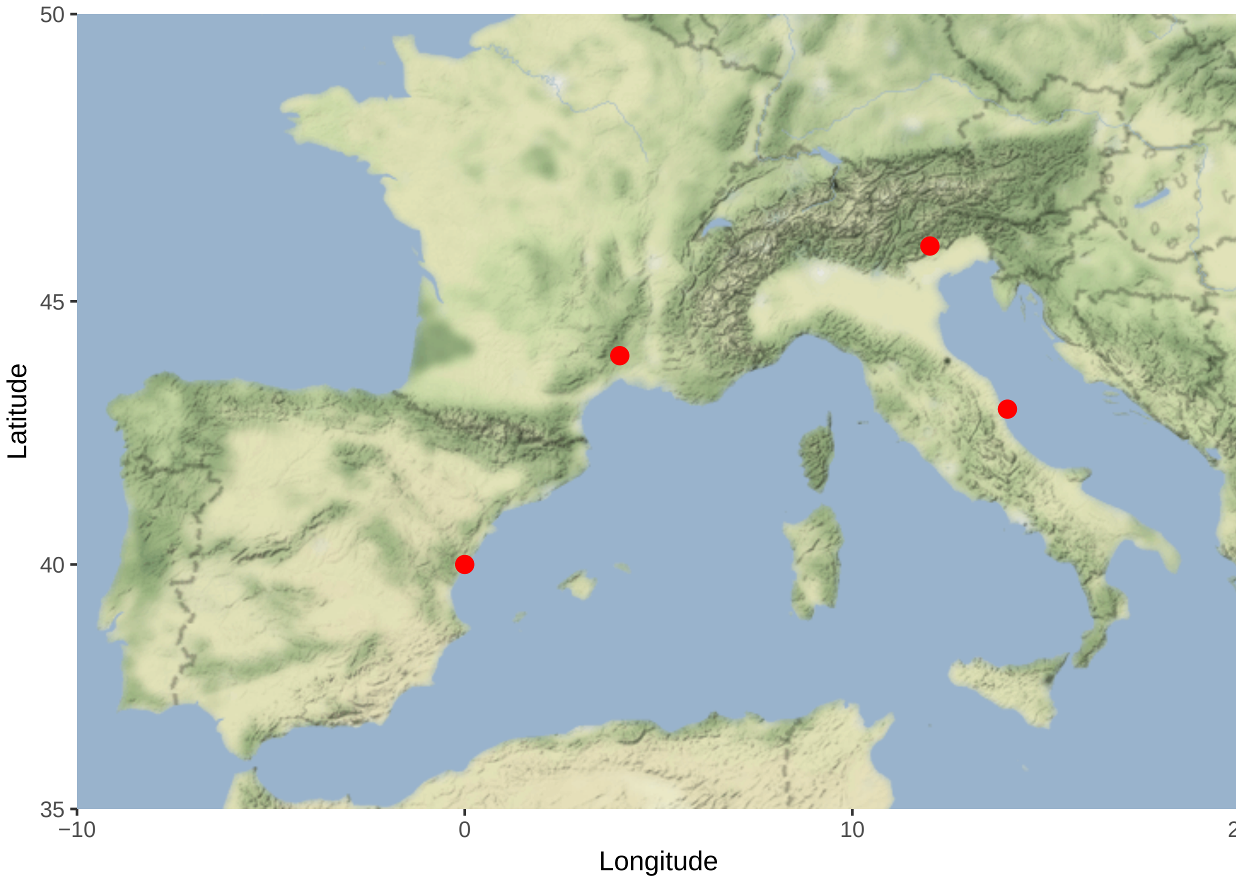
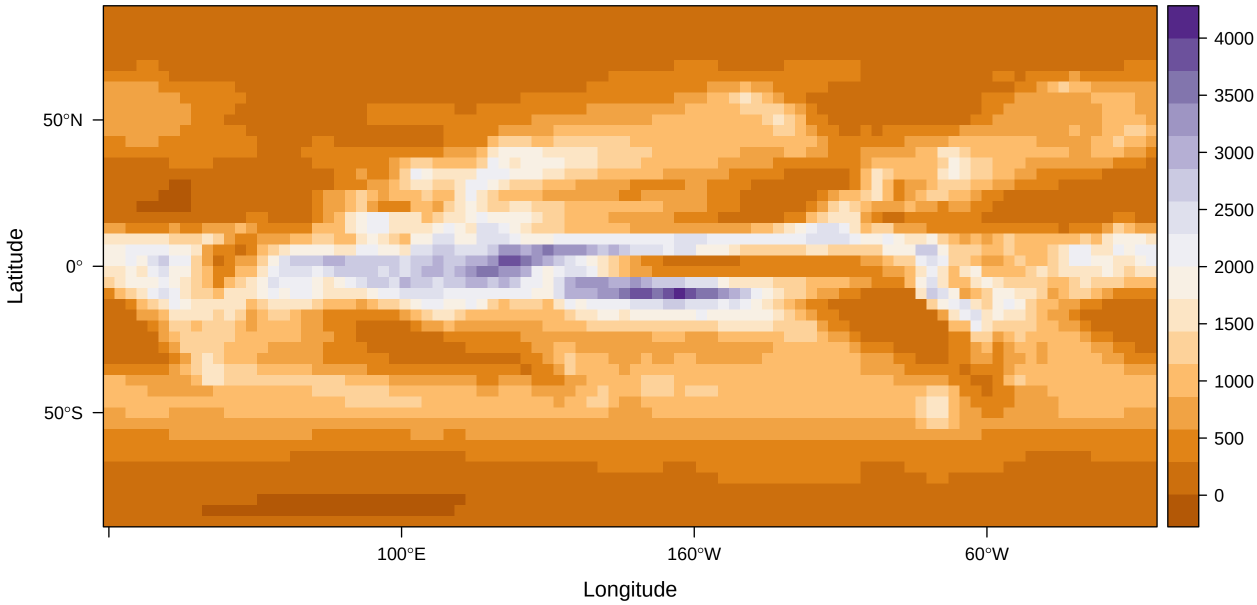
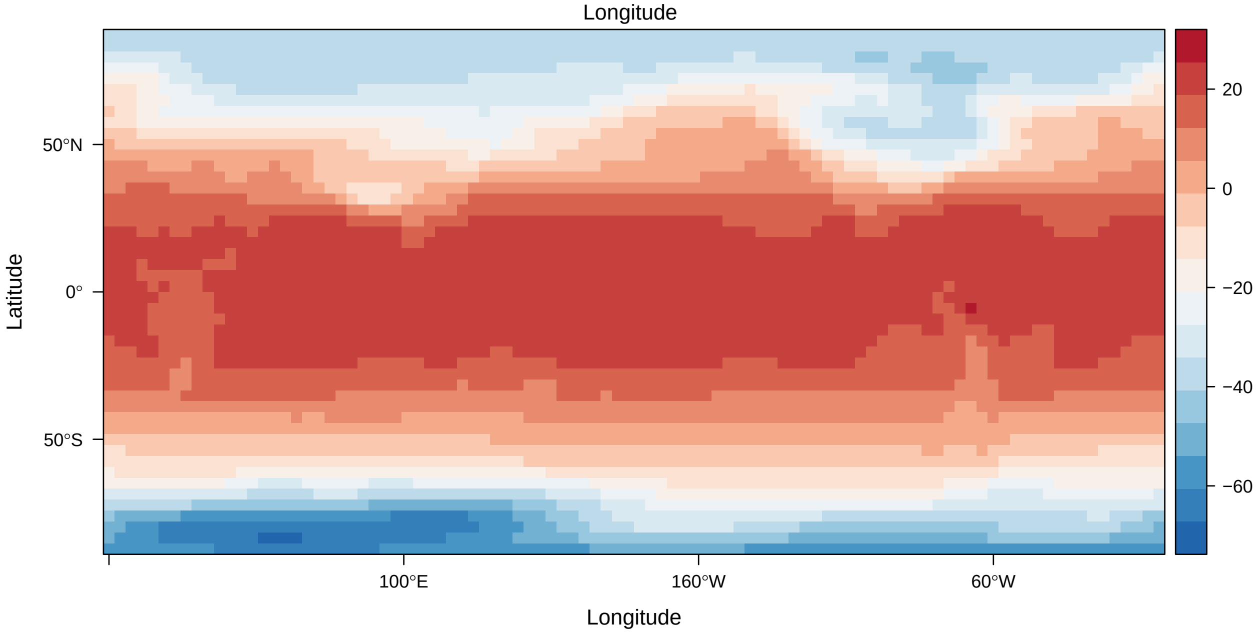



Figure 1: Locations of 3 sample points.

### TraCE-21k Import and Preprocessing

First, import data from the TraCE-21k paleoclimate simulation. Then extract temperature and precipitation values at three locations in the west Mediterranean. Use the *brick* function from **raster** to import decadal averages from the simulation. Put the coordinates for the three locations in a matrix, and use that matrix to and **raster's** *extract* function to get the values from the climate model brick. Convert the precipitation values to mm/year and temperature values to degrees Celsius. Finally, name the columns for each region appropriately.




Now pull all the TraCE data into one data frame, with one row per year, and one column per variable/location combination. First *rbind* the two sets of TraCE data and *transpose* the results, turning the 6 rows into 6 columns. Add a column for the Year (in ka BP), and use to select only the entries earlier than 6,000 BP.

```
# Preprocessing of raw data, skip if using included .RDS file
trace.dat <- rbind(
  brick('trace.01-36.22000BP.cam2.TREFHT.22000BP_decavg_400BCE.nc') %>%
    raster::extract(samp.pts) %>%
    subtract(273.15), # convert from kelvin to C
```

```r
  brick('trace.01-36.22000BP.cam2.PRECT.22000BP_decavg_400BCE.nc') %>%
    raster::extract(samp.pts) %>% # extract data at these coordinates
    multiply_by(3.154e+10)) %>% # convert to mm/year
  t %>% # transpose
  as.data.frame %>%
  set_colnames(c('tmp,Southwest', 'tmp,North Central', 'tmp,Northeast', 'tmp,Southeast',
                 'prc,Southwest', 'prc,North Central', 'prc,Northeast', 'prc,Southeast')) %>%
  rownames_to_column('Year') %>%
  mutate(Year = as.numeric(substring(Year, 3))) %>%
  filter(Year > 6) # get all the decades up to 6ka BP

saveRDS(trace.dat, 'trace_data_points.rds')
```

```r
trace.dat <- readRDS("trace_data_points.rds")
```

## Trend Analysis

Let's use the **EMD** package to calculate trend lines using the empirical mode decomposition approach.

Now organize the temperature and precipitation data to make plotting easier using functions from **tidyr**, along with **EMD** for the trends.

```r
trace.plot <- trace.dat %>% gather(key, value, -Year) %>% separate(key, c("Variable",
    "Region"), ",") %>% mutate(Region = factor(Region, levels = c("Southwest",
    "North Central", "Northeast", "Southeast")), Variable = ifelse(Variable ==
    "tmp", "Temperature (°C)", "Precipitation (mm)"))

emd.res <- function(x) emd(x, boundary = "wave")$residue
trace.emd <- trace.dat %>% mutate_at(vars(-Year), emd.res) %>% gather(key, value,
    -Year) %>% separate(key, c("Variable", "Region"), ",") %>% mutate(Region = factor(Region,
    levels = c("Southwest", "North Central", "Northeast", "Southeast")), Variable = ifelse(Vari
    "tmp", "Temperature (°C)", "Precipitation (mm)"))
```

## Plotting

Plot everything with **ggplot2**.

```r
ggplot(data = trace.plot, aes(x = Year, y = value)) + facet_grid(Variable ~
    Region, switch = "y", scale = "free_y") + geom_vline(xintercept = c(22,
    19, 14, 10, 6), lty = 2) + geom_point(aes(color = Variable), alpha = 0.3) +
    geom_line(data = trace.emd, size = 1.2, color = "black", alpha = 0.8) +
    scale_x_reverse(breaks = seq(6, 22, 4)) + labs(x = "1,000 Years BP", y = "") +
    guides(color = "none") + theme_bw(base_size = 20) + theme(strip.background = element_blank(
```

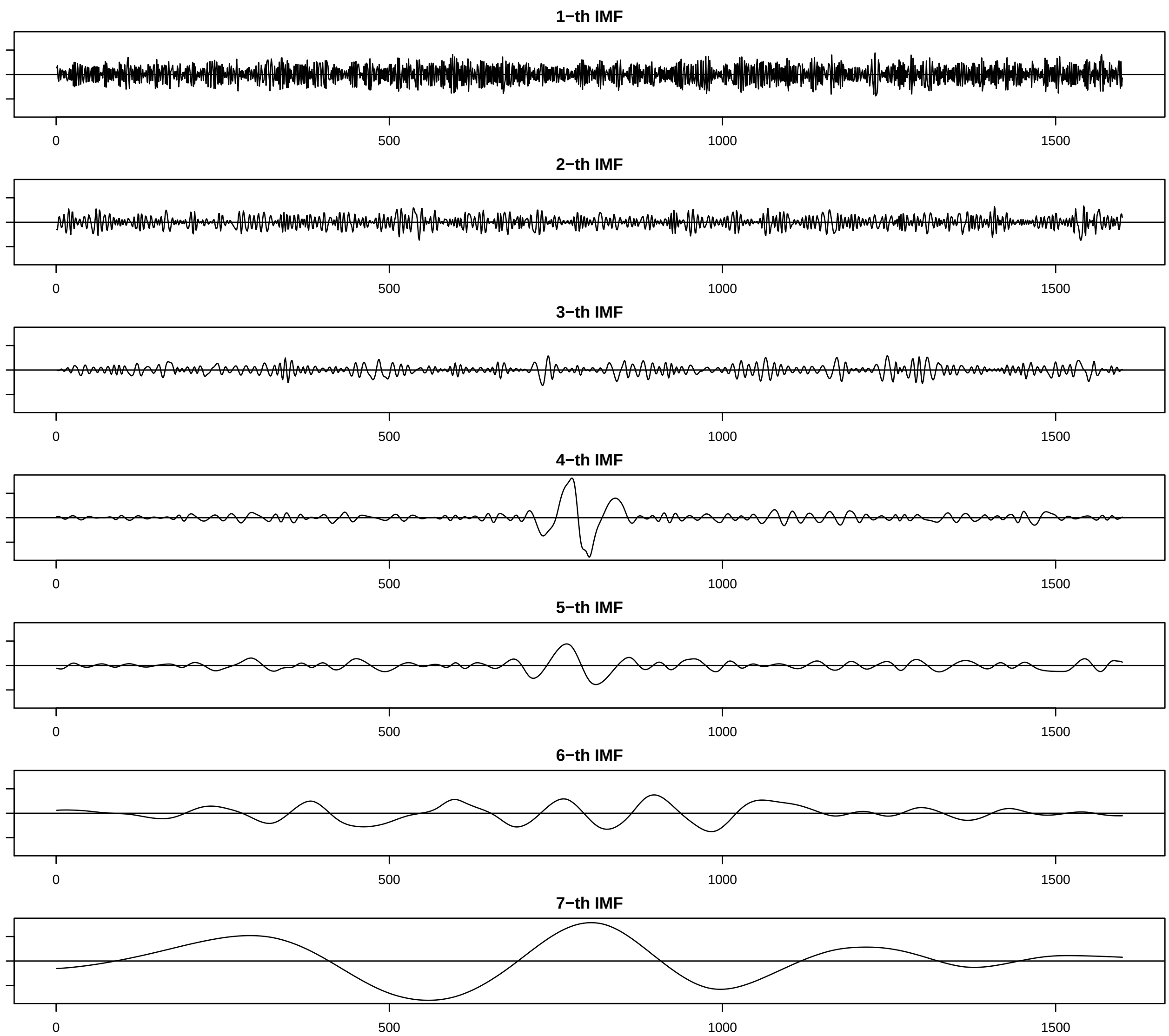
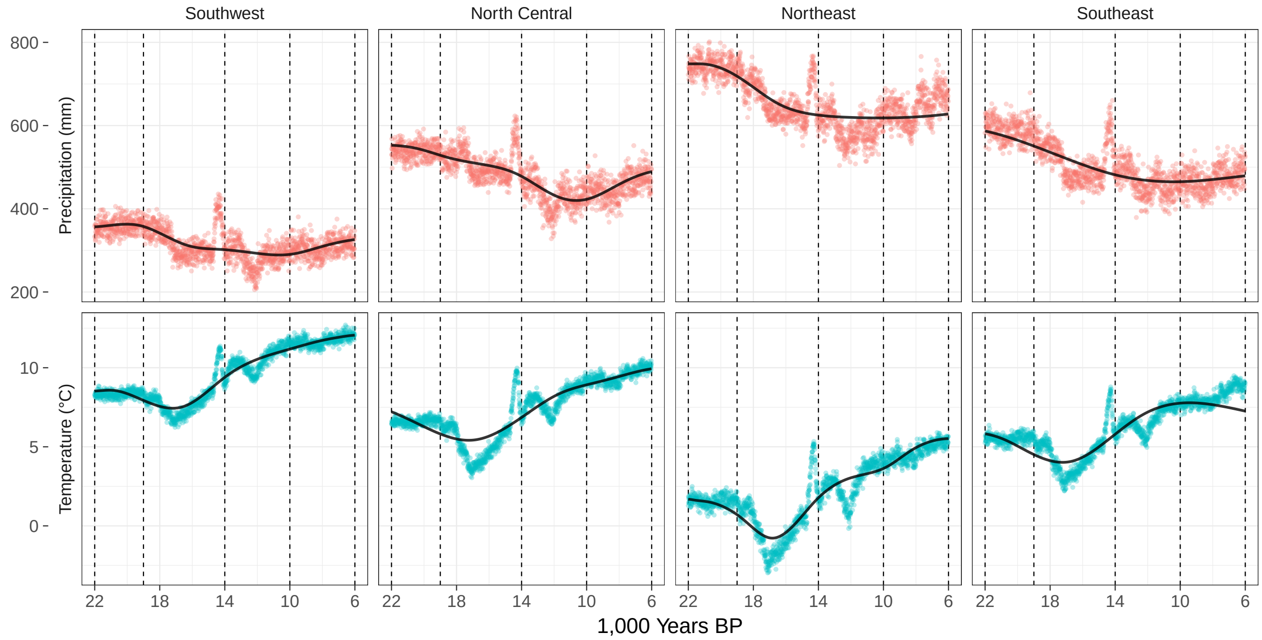



Figure 2: Empirical mode decomposition example




## Variance

Calculate the detrended variances.

```
emd.dat <- trace.dat %>% mutate_at(vars(-Year), emd.res)
```

```
(trace.dat - emd.dat) %>% select(-Year) %>% cbind(Year = trace.dat$Year, .) %>%
    mutate(Period = cut(Year, c(22, 19, 14, 10, 6))) %>% group_by(Period) %>%
    summarise_each(funs(var)) %>% select(-Year) %>% print(width = Inf)
```

```
## `summarise_each()` is deprecated.
## Use `summarise_all()`, `summarise_at()` or `summarise_if()` instead.
## To map `funs` over all variables, use `summarise_all()`
```

```
## # A tibble: 4 x 9
##     Period `tmp,Southwest` `tmp,North Central` `tmp,Northeast` `tmp,Southeast` `prc,Southwest
##     <fctr>           <dbl>               <dbl>           <dbl>           <dbl>           <dbl:
## 1   (6,10]      0.08704209           0.0922012       0.1853124       0.4481085        488.1919
## 2  (10,14]      0.27096878           0.3907428       0.8121861       0.3543841        772.4411
## 3  (14,19]      0.40989953           1.4185010       1.3565404       0.9364233       1317.6964
## 4  (19,22]      0.10400222           0.2620887       0.2029233       0.2716865        313.4281
```

How do these compare to the overall regional variance? First define our region of interest.

```
bbox <- extent(c(-10, 20, 35, 47))
```

Repeat the above process for regional averages.

```
# Preprocessing of raw data, skip if using included .RDS file
trace.reg.avg <- rbind(
```

```
  brick('trace.01-36.22000BP.cam2.TREFHT.22000BP_decavg_400BCE.nc') %>%
    raster::extract(bbox, fun = mean) %>%
    subtract(273.15), # convert from kelvin to C
  brick('trace.01-36.22000BP.cam2.PRECT.22000BP_decavg_400BCE.nc') %>%
    raster::extract(bbox, fun = mean) %>%
    multiply_by(3.154e+10)) %>% # convert to mm/year
  t %>% # transpose
  as.data.frame %>%
  set_colnames(c('tmp,StudyArea', 'prc,StudyArea')) %>%
  rownames_to_column('Year') %>%
  mutate(Year = as.numeric(substring(Year, 3))) %>%
  filter(Year > 6)
saveRDS(trace.reg.avg, 'trace_data_regional.rds')

trace.reg.avg <- readRDS(”trace_data_regional.rds”)

emd.reg.avg <- trace.reg.avg %>% mutate_at(vars(-Year), emd.res)

(trace.dat - emd.dat) %>% select(-Year) %>% cbind(Year = trace.dat$Year, .) %>%
    mutate(Period = cut(Year, c(22, 19, 14, 10, 6))) %>% group_by(Period) %>%
    summarise_each(funs(var)) %>% select(-Year) %>% subtract(((trace.reg.avg -
    emd.reg.avg) %>% magrittr::extract(c(2, 2, 2, 2, 3, 3, 3, 3)) %>% cbind(Year = trace.dat$Ye
    .) %>% mutate(Period = cut(Year, c(22, 19, 14, 10, 6))) %>% group_by(Period) %>%
    summarise_each(funs(var)) %>% select(-Year))) %>% print(width = Inf)

## `summarise_each()` is deprecated.
## Use `summarise_all()`, `summarise_at()` or `summarise_if()` instead.
## To map `funs` over all variables, use `summarise_all()`
## `summarise_each()` is deprecated.
## Use `summarise_all()`, `summarise_at()` or `summarise_if()` instead.
## To map `funs` over all variables, use `summarise_all()`

##   Period tmp,Southwest tmp,North Central tmp,Northeast tmp,Southeast
## 1     NA   -0.03803870       -0.03287960    0.06023160    0.32302766
## 2     NA   -0.16399338       -0.04421932    0.37722393   -0.08057806
## 3     NA   -0.58309403        0.42550744    0.36354680   -0.05657024
## 4     NA   -0.06821138        0.08987510    0.03070968    0.09947292
##   prc,Southwest prc,North Central prc,Northeast prc,Southeast
## 1      127.1965          417.4842      869.1004      378.1488
## 2      288.8544          531.8784      707.5902      453.3773
## 3      109.8887          101.0366      359.6112      813.7744
## 4      102.7977          180.9033      321.0029      438.5091
```

# Spatial Patterns: PMIP3 Ensemble

## Data Preprocessing

First change the study area to all of Europe and the Mediterranean.

```
bbox <- extent(c(-10, 45, 30, 50))
```

Import observed precipitation and temperature normals.

```
# Preprocessing of raw data, skip if using included .RDS files
tmean.obs <- list.files("MOD11C3v5.0-CHIRPSv2.0_MONTHLY_03m/meantemp", full.names = T) %>%
    stack %>% set_names(month.name) %>% crop(bbox)
tmean.obs[tmean.obs == -9999] <- NA

p.obs <- list.files("MOD11C3v5.0-CHIRPSv2.0_MONTHLY_03m/precip", full.names = T) %>%
    stack %>% set_names(month.name) %>% crop(bbox)
p.obs[p.obs == -9999] <- NA

saveRDS(tmean.obs, "tmean_obs.rds")
saveRDS(p.obs, "prec_obs.rds")
```

```
tmean.obs <- readRDS("tmean_obs.rds")
p.obs <- readRDS("prec_obs.rds")
```

Import and reproject SRTM DEM.

```
# Preprocessing of raw data, skip if using included .RDS files
elev <- raster("SRTM_1km.tif") %>% projectRaster(p.obs[[1]]) %>% mask(p.obs[[1]])

saveRDS(elev, "elev.rds")
```

```
elev <- readRDS("elev.rds")
```

Use the DEM to calculate a diffusive continentality (DCO) map, with distance to the sea in km.

```
dco <- elev %>%
  reclassify(c(-Inf, Inf, NA, NA, NA, 1)) %>% # reverse NA and non-NA cells
  distance(doEdge = T) %>% # calculate distances
  mask(elev) %>% # mask out ocean cells
  divide_by(1000) # convert to km
```

Import and preprocess of ERA-interim (ECMWF) reanalysis data, monthly means of daily means, 1979-2010.

```
# Preprocessing of raw data, skip if using included .RDS files
processECMWF <- function(file, var) {
    brick(file, varname = var) %>% stackApply(indices = 1:12, fun = mean) %>%
        rotate %>% set_names(month.name) %>% projectRaster(p.obs[[1]]) %>% mask(p.obs[[1]])
```

```
}

tcw <- processECMWF("ecmwf_surface.nc", "tcw")
msl <- processECMWF("ecmwf_surface.nc", "msl")
t2m <- processECMWF("ecmwf_surface.nc", "t2m")
lsp <- processECMWF("ECMWF Precip.nc", "lsp")
cp <- processECMWF("ECMWF Precip.nc", "cp")

saveRDS(tcw, "tcw_reanalysis.rds")
saveRDS(msl, "msl_reanalysis.rds")
saveRDS(t2m, "t2m_reanalysis.rds")
saveRDS(lsp, "lsp_reanalysis.rds")
saveRDS(cp, "cp_reanalysis.rds")
```

```
tcw <- readRDS("tcw_reanalysis.rds")
msl <- readRDS("msl_reanalysis.rds")
t2m <- readRDS("t2m_reanalysis.rds")
lsp <- readRDS("lsp_reanalysis.rds")
cp <- readRDS("cp_reanalysis.rds")
```

Put all the predictor and response variables together, month by month.

```
cal.vars <- sapply(1:12, function(x) {
    brick(tmean.obs[[x]], p.obs[[x]], msl[[x]], t2m[[x]], tcw[[x]], lsp[[x]],
        cp[[x]], elev, dco) %>% setNames(c("tmean.obs", "p.obs", "msl", "t2m",
        "tcw", "lsp", "cp", "elev", "dco"))
})
```

Sample the variables at random points, weighting for latitude

```
cal.data <- lapply(cal.vars, function(x) (raster::extract(x, randomPoints(elev,
    20000)) %>% data.frame)) %>% do.call(rbind, .)
```

## Model Fitting

Use *mgcv* to fit gams to the combined calibration data. Model precipitation occurence and amounts separately.

```
tmean.gam <- gam(tmean.obs ~ s(t2m, bs = "cr") + s(msl, bs = "cr") + s(elev,
    bs = "cr"), method = "REML", data = cal.data)

p.occur.gam <- gam(factor(p.obs >= 0.1) ~ s(t2m) + s(cp), family = binomial,
    method = "REML", data = cal.data)

prcp.gam <- bam(p.obs ~ s(msl, bs = "cr") + s(tcw, bs = "cr") + s(lsp, bs = "cr") +
    s(cp, bs = "cr") + s(elev, bs = "cr") + s(dco, bs = "cr"), family = Gamma(link = "log"),
    method = "REML", data = cal.data[cal.data$p.obs >= 0.1, ])
```

### Predictions

Write a function to import, process, and generate a monthly average ensemble from PMIP3 data. This function assumes the PMIP3 data are organized with a "PMIP3 Data > Period > Variable > Model" directory structure.

```
getEns <- function(period, variable) {
    var.dir <- paste0("PMIP3 Data/", period, "/", variable)
    files.in <- list.files(var.dir, full.names = T)

    sapply(files.in, function(x) {
        brick(x) %>% rotate %>% projectRaster(elev)
    }) %>% brick %>% stackApply(indices = 1:12, fun = mean)
}
```

#### Mid Holocene

Use this function to import all the necessary variables.

```
# Preprocessing of raw data, skip if using included .RDS files
t2m <- getEns("MH", "tas")
msl <- getEns("MH", "psl")
cp <- getEns("MH", "prc") %>% multiply_by(86.4)
lsp <- (getEns("MH", "pr") %>% multiply_by(86.4)) - cp
tcw <- getEns("MH", "clivi") + getEns("MH", "clwvi") + getEns("MH", "prw")

saveRDS(t2m, "t2m_mh.rds")
saveRDS(msl, "msl_mh.rds")
saveRDS(cp, "cp_mh.rds")
saveRDS(lsp, "lsp_mh.rds")
saveRDS(tcw, "tcw_mh.rds")
```

```
t2m <- readRDS("t2m_mh.rds")
msl <- readRDS("msl_mh.rds")
cp <- readRDS("cp_mh.rds")
lsp <- readRDS("lsp_mh.rds")
tcw <- readRDS("tcw_mh.rds")
```

Generate a single prediction variable set.

```
pred.vars.mh <- sapply(1:12, function(x) {
    brick(t2m[[x]], msl[[x]], tcw[[x]], cp[[x]], lsp[[x]], crop(elev, t2m),
        crop(dco, t2m)) %>% setNames(c("t2m", "msl", "tcw", "cp", "lsp", "elev",
        "dco"))
})
```

Make predictions for the Mid Holocene.

```
tmean.mh <- sapply(1:12, function(x) {
    predict(pred.vars.mh[[x]], tmean.gam)
}) %>% brick

prec.occur.mh <- sapply(1:12, function(x) {
    predict(pred.vars.mh[[x]], p.occur.gam, type = "response")
}) %>% brick %>% is_weakly_greater_than(0.5)

prec.mh <- sapply(1:12, function(x) {
    predict(pred.vars.mh[[x]], prcp.gam, type = "response")
}) %>% brick %>% mask(prec.occur.mh, maskvalue = 0, updatevalue = 0)
```

### LGM

Repeat for the LGM

```
# Preprocessing of raw data, skip if using included .RDS files
t2m <- getEns("LGM", "tas")
msl <- getEns("LGM", "psl")
cp <- getEns("LGM", "prc") %>% multiply_by(86.4)
lsp <- (getEns("LGM", "pr") %>% multiply_by(86.4)) - cp
tcw <- getEns("LGM", "clivi") + getEns("LGM", "clwvi") + getEns("LGM", "prw")

saveRDS(t2m, "t2m_lgm.rds")
saveRDS(msl, "msl_lgm.rds")
saveRDS(cp, "cp_lgm.rds")
saveRDS(lsp, "lsp_lgm.rds")
saveRDS(tcw, "tcw_lgm.rds")

t2m <- readRDS("t2m_lgm.rds")
msl <- readRDS("msl_lgm.rds")
cp <- readRDS("cp_lgm.rds")
lsp <- readRDS("lsp_lgm.rds")
tcw <- readRDS("tcw_lgm.rds")

pred.vars.lgm <- sapply(1:12, function(x) {
    brick(t2m[[x]], msl[[x]], tcw[[x]], cp[[x]], lsp[[x]], crop(elev, t2m),
        crop(dco, t2m)) %>% setNames(c("t2m", "msl", "tcw", "cp", "lsp", "elev",
        "dco"))
})

tmean.lgm <- sapply(1:12, function(x) {
    predict(pred.vars.lgm[[x]], tmean.gam)
}) %>% brick

prec.occur.lgm <- sapply(1:12, function(x) {
```

```
    predict(pred.vars.lgm[[x]], p.occur.gam, type = "response")
}) %>% brick %>% is_weakly_greater_than(0.5)

prec.lgm <- sapply(1:12, function(x) {
    predict(pred.vars.lgm[[x]], prcp.gam, type = "response")
}) %>% brick %>% mask(prec.occur.lgm, maskvalue = 0, updatevalue = 0)
```

## Analysis of Spatial Patterns

Use the downscaled ensemble data to estimate how the spatial patterns of climate variability have changed over time, and to test for consistency with the transient TraCE simulation.

First crop the downscaled data to the West Mediterranean.

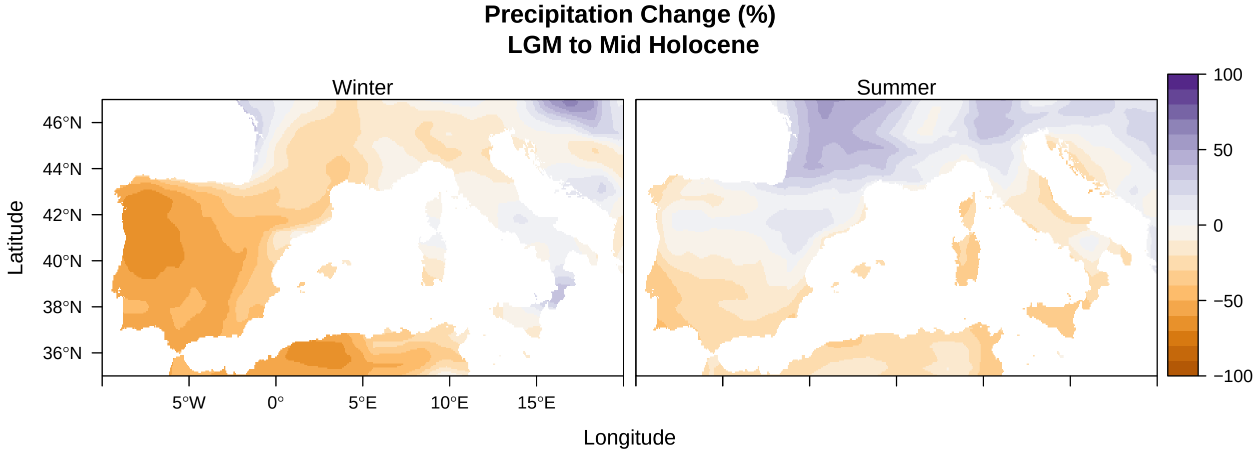
```
bbox <- extent(c(-10, 20, 35, 47))

lgm.prc <- prec.lgm %>% crop(bbox)
mh.prc <- prec.mh %>% crop(bbox)

lgm.tmp <- tmean.lgm %>% crop(bbox)
mh.tmp <- tmean.mh %>% crop(bbox)
```

Calculate changes in seasonal precipitation and temperature.

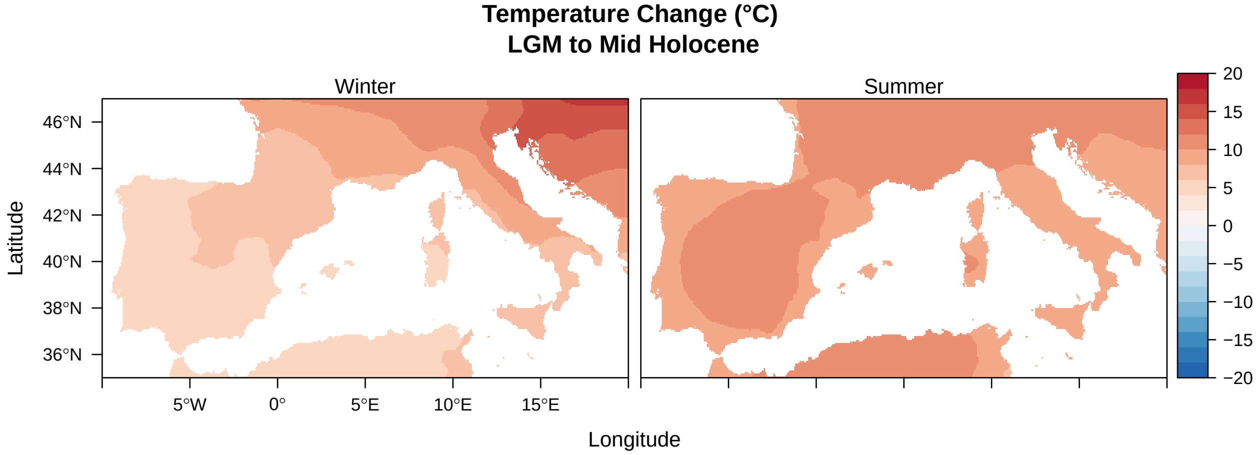
```
bySeason <- function(x, season, var) {
    if (season == "djf") {
        ids <- c(1, 2, 12)
    }
    if (season == "jja") {
        ids <- c(6, 7, 8)
    }

    if (var == "tmp")
        return(mean(x[[ids]]))
    if (var == "prc")
        return(sum(x[[ids]]))
}

prc.change.map.percent <- brick(c((bySeason(mh.prc, "djf", "prc") - bySeason(lgm.prc,
    "djf", "prc")) * 100/bySeason(lgm.prc, "djf", "prc"), (bySeason(mh.prc,
    "jja", "prc") - bySeason(lgm.prc, "jja", "prc")) * 100/bySeason(lgm.prc,
    "jja", "prc")))

tmp.change.map <- brick(c(bySeason(mh.tmp, "djf", "tmp") - bySeason(lgm.tmp,
    "djf", "tmp"), bySeason(mh.tmp, "jja", "tmp") - bySeason(lgm.tmp, "jja",
    "tmp")))
```

Plot the results.

```
levelplot(prc.change.map.percent, margin = F, names.attr = c("Winter", "Summer"),
    main = "Precipitation Change (%)\n LGM to Mid Holocene", par.settings = PuOrTheme(),
    at = seq(-100, 100, 10))
```




```
levelplot(tmp.change.map, margin = F, names.attr = c("Winter", "Summer"), main = "Temperature C
    par.settings = BuRdTheme(), at = seq(-20, 20, 2))
```

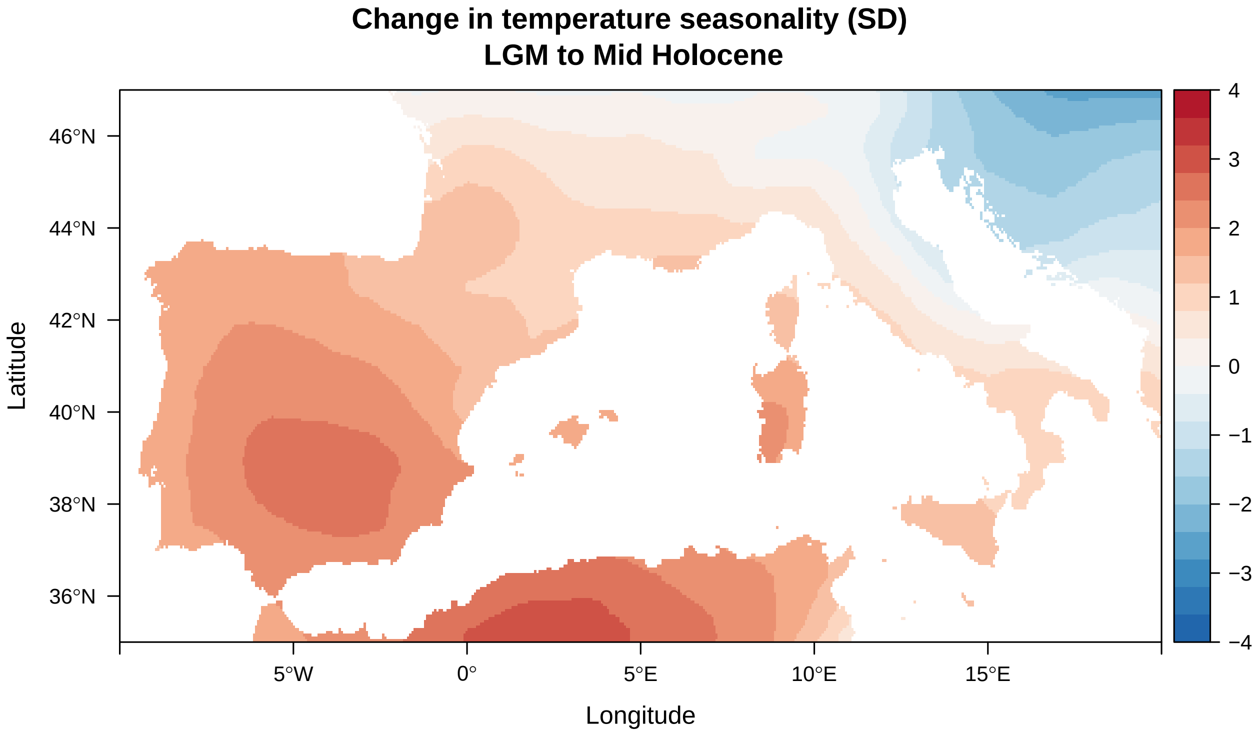



Now we can calculate changes in seasonality. For temperature, this is just the standard deviation of all 12 monthly averages. For precipitation, we will use the coefficient of variation.

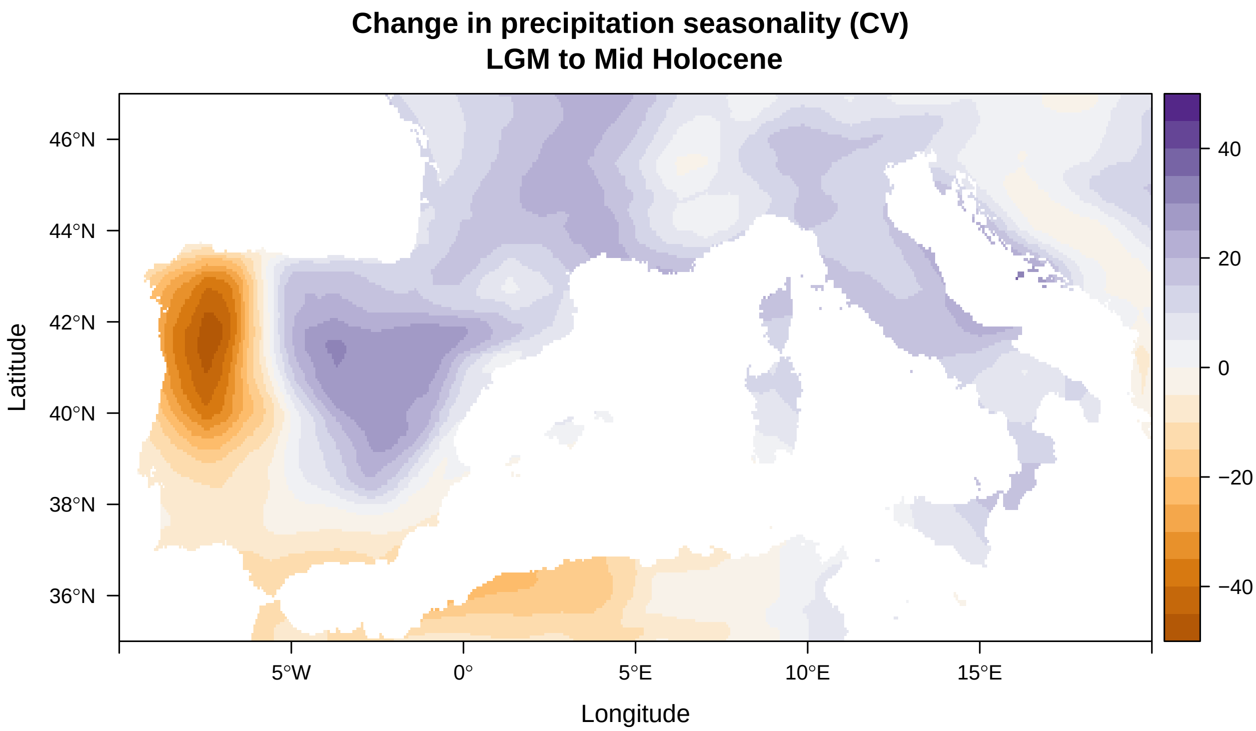
```
tmp.seasonality <- calc(mh.tmp, sd) - calc(lgm.tmp, sd)
prc.seasonality <- cv(mh.prc) - cv(lgm.prc)
```

Plot the results.

```
levelplot(tmp.seasonality, margin = F, main = "Change in temperature seasonality (SD)\n LGM to M
    par.settings = BuRdTheme(), at = seq(-4, 4, 0.4))
```




```
levelplot(prc.seasonality, margin = F, main = "Change in precipitation seasonality (CV)\n LGM t
    par.settings = PuOrTheme(), at = seq(-50, 50, 5))
```



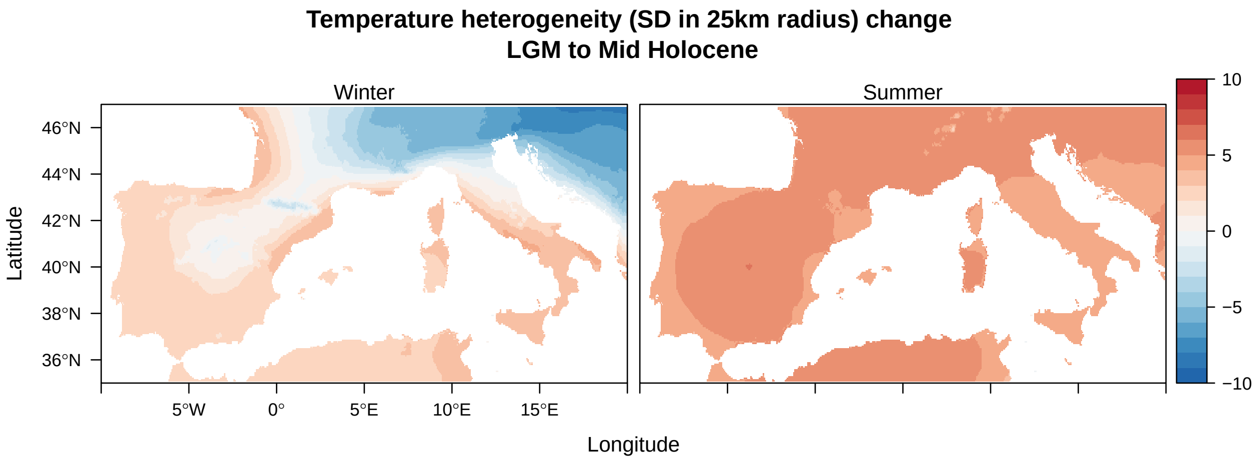

What about changes in spatial hetergeneity? Define a 5x5 weights matrix within which to sample the standard deviation of the climate maps.

```
wts <- matrix(c(0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0,
```

```
    0, 0, 1, 0, 0), nrow = 5)
```

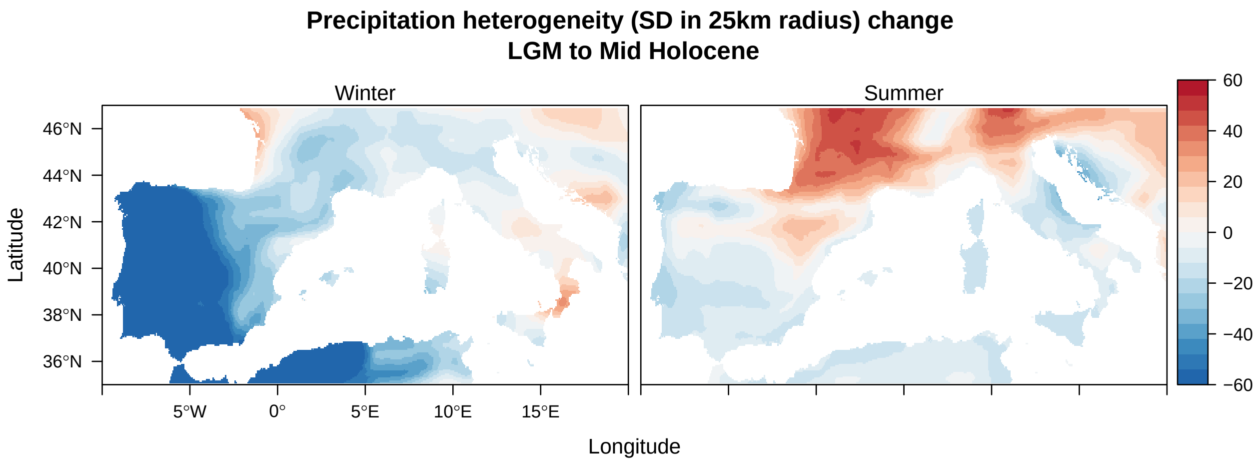
Temperature

```
tmp.hetero <- brick(c(bySeason(mh.tmp, "djf", "tmp") %>% focal(w = wts, sd,
    na.rm = T) %>% subtract(bySeason(lgm.tmp, "djf", "tmp") %>% focal(w = wts,
    sd, na.rm = T)), bySeason(mh.tmp, "jja", "tmp") %>% focal(w = wts, sd, na.rm = T) %>%
    subtract(bySeason(lgm.tmp, "jja", "tmp") %>% focal(w = wts, sd, na.rm = T)))) %>%
    mask(mh.tmp[[1]])  # clip buffer added by window

levelplot(tmp.hetero, margin = F, names.attr = c("Winter", "Summer"), main = "Temperature hetero
    par.settings = BuRdTheme(), at = seq(-10, 10, 1))
```




Same for precipitaiton.

```
prc.hetero.sd <- brick(c(bySeason(mh.prc, "djf", "prc") %>% focal(w = wts, sd,
    na.rm = T) %>% subtract(bySeason(lgm.prc, "djf", "prc") %>% focal(w = wts,
    sd, na.rm = T)), bySeason(mh.prc, "jja", "prc") %>% focal(w = wts, sd, na.rm = T) %>%
    subtract(bySeason(lgm.prc, "jja", "prc") %>% focal(w = wts, sd, na.rm = T)))) %>%
    mask(mh.prc[[1]])  # clip buffer added by window function

# capped at -60 to reveal patterns
prc.hetero.sd[prc.hetero.sd < -60] <- -60
levelplot(prc.hetero.sd, margin = F, names.attr = c("Winter", "Summer"), main = "Precipitation 
    par.settings = BuRdTheme(), at = seq(-60, 60, 6))
```


Precipitation heterogeneity (SD in 25km radius) change
LGM to Mid Holocene
Winter
Summer
46°N
44°N
42°N
40°N
38°N
36°N
Latitude
5°W
0°
5°E
10°E
15°E
Longitude
60
40
20
0
−20
−40
−60