Conversion and analysis of historical air pressure data

Jonas Bhend

August 21, 2014

Contents

1 Installation and preparation

All the code necessary to convert and analyse pressure readings is available on github.

The main repository contains the following sub-directories.

- `docs`: documentation on the pressure conversion and analysis (this manual)
- `src`: the shell scripts to compute 30-year climatologies of 20th century reanalysis temperatures and the R scripts to read in all available stations in the repository
- `compact_data`: Original records (.xls files) converted to the standard compact format
- `orig_data`: Original records not converted to the standard format or links to the original records where these can be used directly

In addition, the repository contains a README file, the spreadsheet containing metadata on the pressure inventory (Working_inventory_pressure_1815-17.xls), and a template for organising pressure data in compact form for automatic processing (Template_for_pressure_data.xlsx).

To get a copy of the repository, please clone the repository from github

```
git clone git@github.com:jonasbhend/pressure.git
```

or download it as a zip file.

To run, inspect and change the various R functions used to convert atmospheric pressure, you also need to install the R package `pressurehelper` from github. To do so from within R, type:

```
install.packages('devtools')
library(devtools)
install_github('jonasbhend/pressurehelper')
```

2 Common data format

The common data format is a table with all available information from the original digitised files. Each line represents a unique observation time and the following list of standard names are used whenever applicable:

- `Latitude, Longitude, Elevation` for the location of the station
- `Location.missing, Elevation.missing` to indicate whether location is known or estimated 0 location or elevation is known 1 location or elevation has been estimated
- `Year, Day, Month, Time` for observation date and time from original record
- `Local.time` for observation time converted to HH:MM
- `Time.missing` to indicate whether time has been estimated 0 time is known 1 time has been estimated (e.g. sunrise)
- `Local.date` Datestring collating the above in local time (YYYY-MM-DD HH:MM:SS)
- `UTC.date` Datestring in UTC (YYYY-MM-DD HH:MM:SS)
- `P, P.1, P.2, P.3` pressure reading in original units
- `P.units` pressure units in original file
- `Tcorr` flag for temperature correction in original record
  - 0 no temperature correction
  - 1 corrected
- `mmHg` barometer readings in mm
- `P.orig` barometer readings in hPa corrected for local gravity
- `QFE` station pressure in hPa reduced to 0 deg. C
- `QFE.flag` temperature correction of station pressure
  - 0 missing value
  - 1 corrected in original record (and sometimes rebased, e.g. from 55 F to 0 C)
  - 2 corrected using temperature at the barometer
  - 3 corrected using in-situ outside air temperature
  - 4 corrected using 20CR climatology
- `QFF` sea level pressure in hPa
- `QFF.flag` temperature used to reduce to sea level
  - 0 missing value

  - 1 QFF available from original record
  - 2 corrected using in-situ outside air temperature
  - 3 corrected using 20CR climatology

- `TP and TA` for temperature at barometer and temperature of outside air

Data in the compact format are read in and converted to the long format using the function `compact2long` from the `pressurehelper` package. The conversion of pressure and temperature data, and the extension of the data frame in the long format to include all available columns specified above is carried out in the `expand_long` function. The individual steps performed to convert observations and expand the data frame are specified in the next section.

3 Processing of pressure data

3.1 Conversion of observation times

Observation times are available in various formats in the original records. All observation times are assumed to refer to local (solar) time.

3.1.1 Missing observation times

In case observation times are missing, but observations have been taken at regular intervals, missing observations times are replaced with the most frequent observation time for this interval (e.g. 9PM for evening observations if 9PM is the most frequent known time for evening observations). If the observation schedule is fully unknown, a local observation time of 2PM is assumed. In both cases, the time flag for these observations is set to missing (i.e. `Time.missing = 1`).

3.1.2 Qualitative observation times

Qualitative observation times are converted to quantitative times based on the lookup table below.

- morning = 8AM
- noon = 12PM
- afternoon = 4PM
- evening = 8PM

3.1.3 Local sunrise and sunset

In cases for which observation times are noted as *sunrise* and *sunset*, the local sunrise and sunset is computed based on the date and latitude of the station using. The following formula is used to estimate local sunrise and sunset

$$H_{sun} = \arccos\left(-\tan\phi \cdot \tan\delta\right) \cdot \frac{24\ h}{2\pi}$$

where H_{sun} is the half-day length in hours, ϕ the latitude of the station and δ the declination of the sun. The declination of the sun δ is computed based on the Julian day and the function `declination` from the R-package `insol`. The local sunrise and sunset time is one half-day length H_{sun} before and after local noon (12PM) respectively.

3.1.4 Translation to UTC

Observation dates and times (see above) from the original record are translated to UTC (for extraction of temperature climatologies and for intercomparison). For the translation to UTC, we use the simple formula based on the longitude of the station.

$$t_{UTC} = t_{loc} - \lambda \cdot \frac{24\ h}{360ž}$$

where λ is the longitude of the station in degrees east, t_{loc} is the local time and t_{UTC} is the UTC time.

Missing observation dates (days, months, and years) are only substituted in cases for which there is no ambiguity (e.g. year and month are only noted on top of page and where they change).

Finally, we assume that the standard Gregorian calendar has been used in all of the records available.

3.2 Conversion of length units

Barometer readings in original units are converted to mm Hg. The conversion factors are detailed below. Generally, only the largest length unit used is indicated, sub-units follow base 12 unless specified.

- English inches: 1 in = 2.54 cm
- French inches: 1 in = 2.707 cm
- Swedish inches: 1 tum = 2.969 cm
- Rijnlands inches: 1 in = 2.62 cm

3.3 Correction of barometer readings to standard conditions

We correct all converted barometer readings in mmHg to standard conditions. This involves a correction for local gravity and a correction for temperature following WMO 2010. Barometer readings that are available in hPa are only corrected for temperature if metadata does not indicate that such a correction has already been performed.

3.3.1 Correction for local gravity

To convert the pressure reading in mmHg to hPa we use the following formula:

$$P_n = \rho \cdot g_{\phi,h} \cdot mmHg \times 10^{-5}$$

where P_n is the absolute pressure in hPa reduced to normal gravity, $\rho = 1.35951 \times 10^4\ kgm^{-3}$ is the density of mercury at 0 degrees C, $g_{\phi,h}$ is the local gravity (see below), and $mmHg$ is the barometer reading in mm. This is equivalent to correcting pressure in hPa for local gravity by using

$$P_n = g_{\phi,h}/g_n \cdot P_0$$

where P_0 is the absolute pressure not reduced to normal gravity and $g_n = 9.80665\ ms\check{}^2$.

Local gravity $g_{\phi,h}$ is estimated based on the latitude ϕ and elevation h (in m above sea level) assuming flat terrain around the station

$$g_{\phi,h} = 9.80620\ ms^{-2} \cdot (1 - 0.0026442 \cdot \cos 2\phi - 0.0000058 \cdot \cos^2 2\phi) - 0.000003086 \cdot h$$

3.3.2 Reduction to 0º C

Pressure readings are converted to 0º C using the following formula

$$P_T = P\ (1 - \gamma \cdot T)$$

where $\gamma = 1.82 \times 10^{-4}\ K^{-1}$ is the thermal expansion coefficient of mercury at 0º C, P is the pressure reading not corrected for temperature, and T is the temperature at the barometer (`TP.orig`). If `TP.orig` is not available, then *in-situ* outside air temperature measurement `TA.orig` is used instead. If no *in-situ* measurements are available, then the closest (in space and time) 30-year climatology of 3-hourly temperatures from the 20$^{\text{th}}$ century reanalysis is used. These climatologies are computed from the years 1871-1900. To further reduce variability, a 11-day moving mean has been applied per timestep, so that the climatology for temperature on January 6, 12:00 UTC, is the average of temperature on January 1-11, 12:00 UTC, for the years 1871-1900.

Where pressure observations have been reduced to some temperature other than 0º C at the time of reading, the pressure readings are rebased to 0º C using above formula.

3.4 Reduction to mean sea level

To further reduce station pressure `QFE` to pressure at sea level `QFF`, we follow the guidelines of WMO 2010, but diverge where necessary. The station pressure P_S (`QFE` in common data format) is reduced to sea level pressure P_0 using

$$P_0 = P_S \cdot \exp\left(\frac{\frac{g(\phi,h)}{R} \cdot h}{T_S + a \cdot \frac{h}{2}}\right)$$

where the $g(\phi, h)$ is the local gravity, $R = 287.05\ Jkg^{-1}K^{-1}$ is the gas constant of dry air, h is the elevation in m, $a = 0.0065\ Km^{-1}$ is the standard lapse rate of the fictitious air column below the station, and T_S is the temperature at the station in K. In contrast to WMO 2010, we do not correct for humidity and instead of using the station elevation in geopotential meters and normal gravity, we use station elevation in m and local gravity. Also, we do not further adjust low-level stations based on the mean annual normal value of virtual temperature.

As for the reduction of pressure readings to 0º C, we use *in-situ* air temperature observations (`TA.orig`) where available and resort to using climatological temperatures from the 20$^{\text{th}}$ century reanalysis (1871-1900) instead. We do not use temperature at the barometer (`TP.orig`) to reduce pressure readings to sea level as this is measured indoors in most cases and may be affected by heating in the room or adjacent rooms.

4 Analysis of pressure data at stations

In the following, the R code used to produce the plots and analyses is displayed to illustrate how to work with the data. First, we load the data converted with `read_and_convert_pressure.R`.

```
library(ggplot2, quietly=TRUE)
library(maps, quietly=TRUE)
library(pressurehelper, quietly=TRUE) ## for computation of distance on globe
```

```
## XLConnect 0.2-7 by Mirai Solutions GmbH
## http://www.mirai-solutions.com ,
## http://miraisolutions.wordpress.com
```

```
## read in all the data (not just 1815-17)
load('../data/all_station_and_travel_pressure_data.Rdata')

## exclude travel data
travelnames <- c('West_coast_of_Corea', 'Interior_of_China', 'England_to_Ceylon')
PP <- PP[-which(names(PP) %in% travelnames)]
PPmerge <- PPmerge[-which(PPmerge$Station %in% travelnames),]
```

Next, we inspect the data visually.

```
ggplot() + geom_boxplot(data=PPmerge, aes(Station, QFE)) +
  coord_flip() + scale_x_discrete(limits=rev(unique(PPmerge$Station)))
```

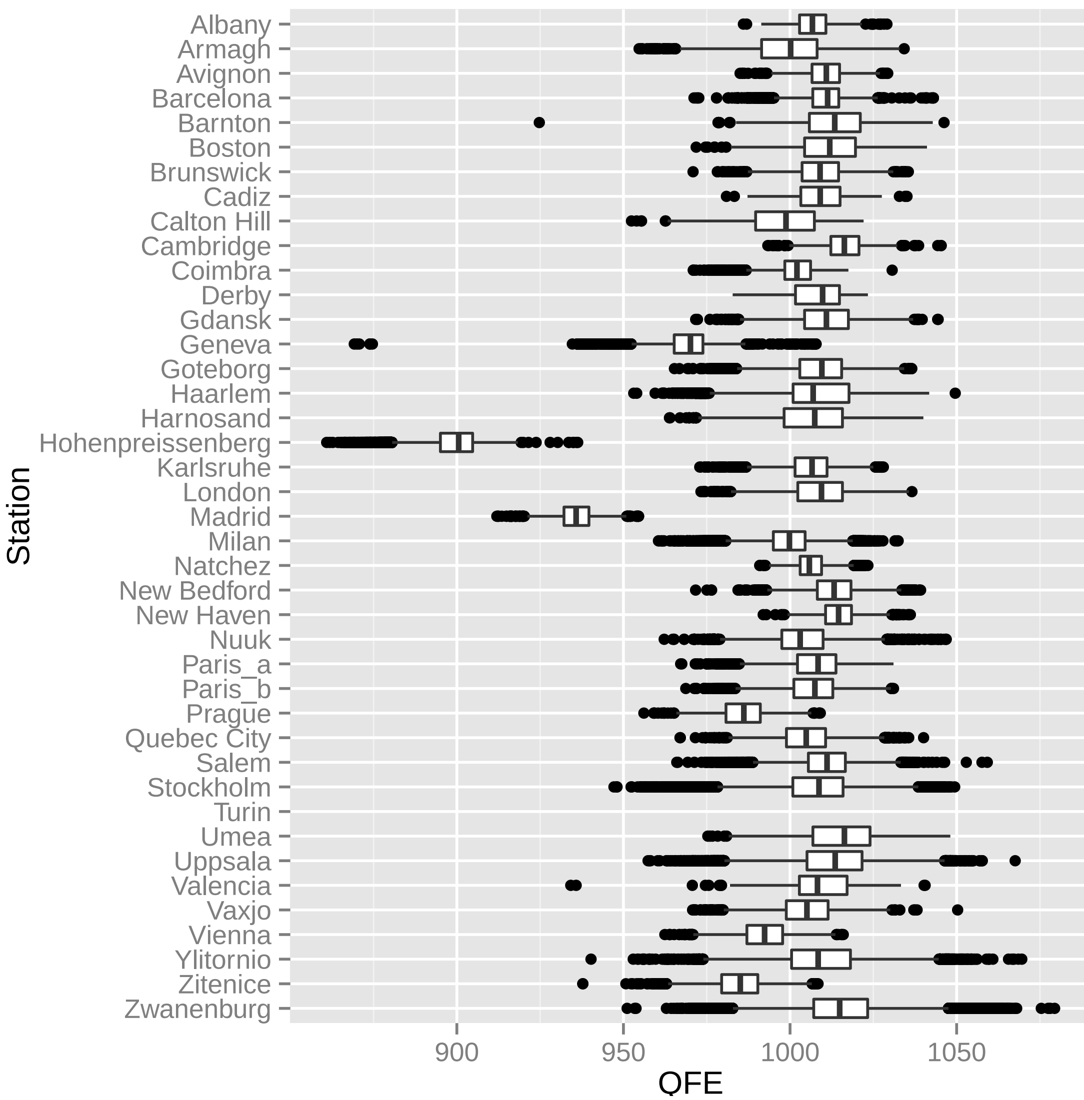

Figure 1: Pressure at station (in hPa) per station in the data set

```
ggplot() + geom_boxplot(data=PPmerge, aes(Station, QFF)) +
  scale_x_discrete(limits=rev(unique(PPmerge$Station))) +
  coord_flip()
```

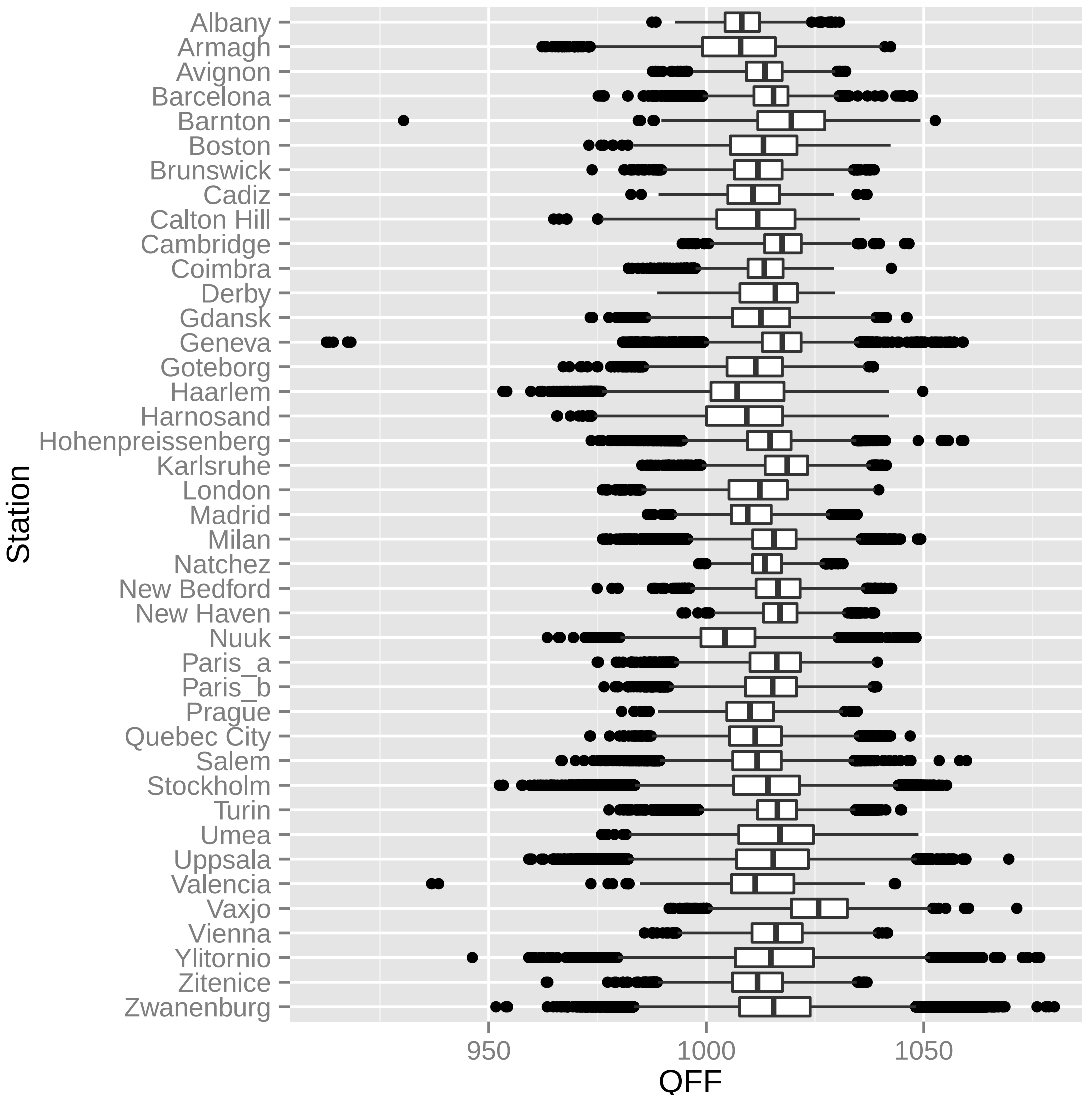

Figure 2: Pressure reduced to sea level (in hPa) per station in the data set

```
ggplot(PPmerge[PPmerge$Year %in% 1815:1817, ], aes(Local.date, QFF)) +
  geom_line() +
  facet_wrap(~ Station, ncol=7) +
  theme(axis.ticks.x=element_blank(), axis.text.x=element_blank()) +
  xlab('Year (1815-1817)')
```

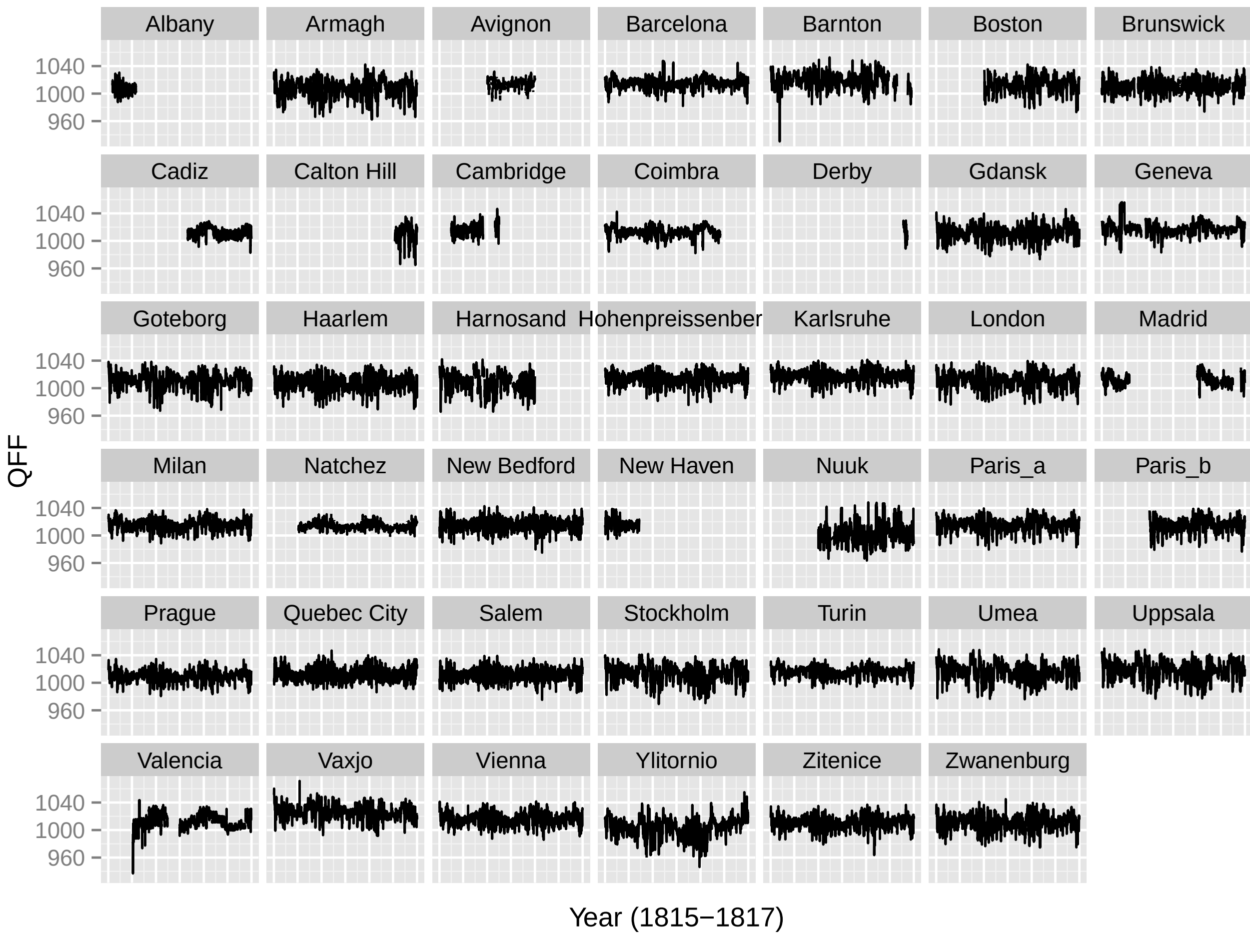

Figure 3: Time series of pressure reduced to sea level (QFF in hPa) per station for 1815-1817

```
## compute trend
PPtrend <- Reduce(rbind, lapply(PP, function(x) {
  xlm <- summary(lm(QFF ~ Local.date, x))$coef * 3600*24 *365
  data.frame(Station=x$Station[1], trend=xlm[2,1],
             lower=xlm[2,1] - 1.96*xlm[2,2],
             upper=xlm[2,1] + 1.96*xlm[2,2])
  }))

ggplot(PPtrend, aes(x=Station, y=trend)) +
  geom_errorbar(mapping=aes(ymin=lower, ymax=upper)) +
  geom_point() +
  scale_x_discrete(limits=rev(unique(PPtrend$Station))) +
  coord_flip()
```

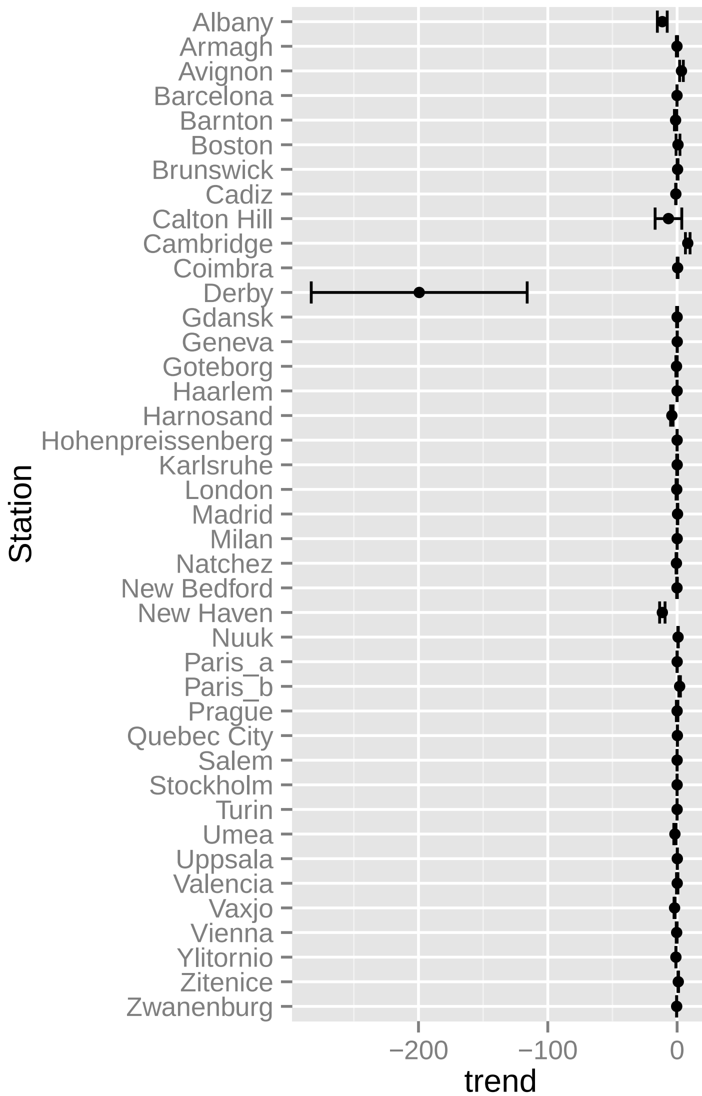

Figure 4: Trend in pressure time series (all available values) in hPa per year with 5-95% confidence interval. This should be zero unless the record is short, or there is a drift in the measurement.

```
monnames <- format(as.Date('1990-01-15') + seq(0,330,30), '%b')

qplot(x=Month, y=QFF,
      group=Station, colour=Station,
      data=PPmerge[PPmerge$Total.values > 1095,],
      stat='summary', fun.y='mean', geom='line') +
  guides(colour=FALSE) +
  scale_x_discrete(name='Month', labels=monnames)
```

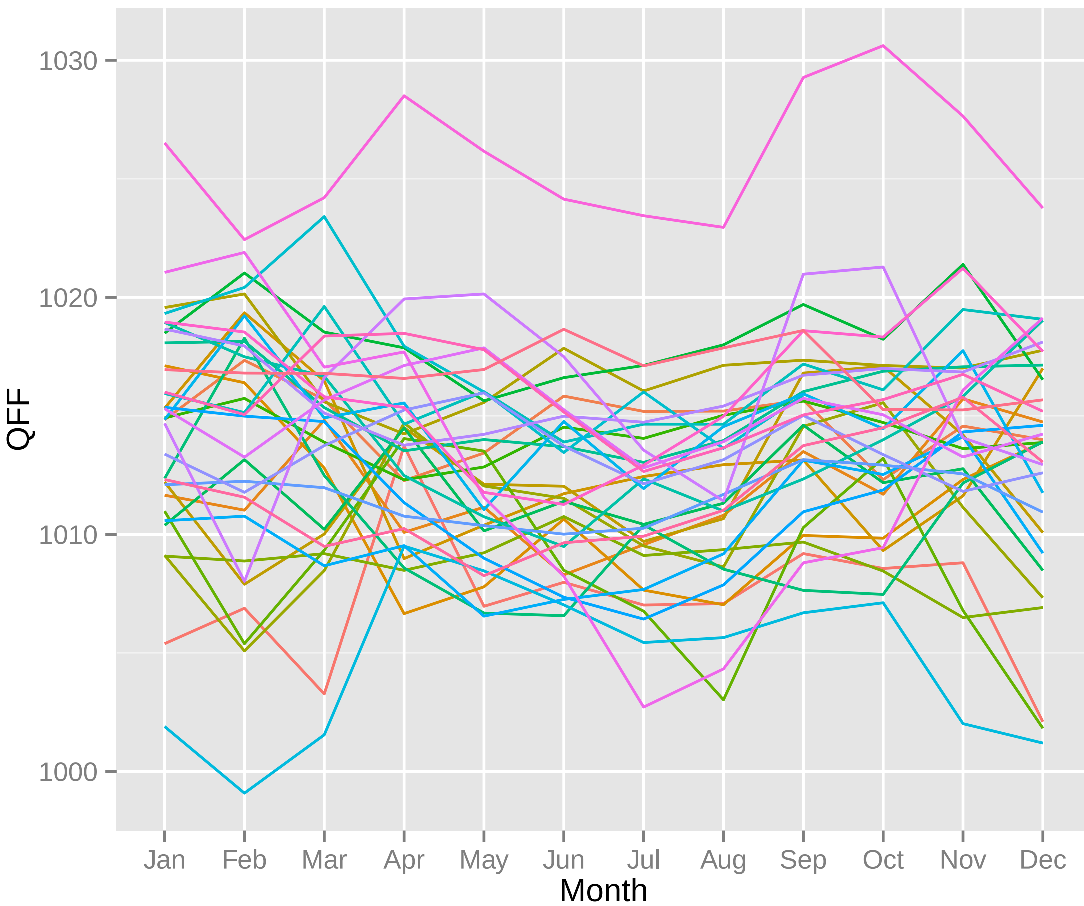

Figure 5: Annual cycle of sea level pressure (QFF, using the complete record for station with at least 1096 values in 1815-17). This is not very helpful and further smoothing might be needed.

```r
dailyfun <- function(x){
  xhour <- as.numeric(format(x$Local.date, '%H'))
  ## subset to morning noon and evening
  xcut <- c('morning', 'noon', 'evening')[cut(xhour, c(0,10,15,24))]
  ## get name to use (QFE or QFF)
  xname <- if (any(!is.na(x$QFE))) 'QFE' else 'QFF'
  xdaily <- tapply(x[[xname]], xcut, mean, na.rm=T)
  if (length(xdaily) > 0) {
    ## resort xdaily to get morning first, evening last
    xdaily <- xdaily[c('morning', 'noon', 'evening')]
    xdaily <- xdaily[!is.na(xdaily)]
    out <- data.frame(Time=names(xdaily),
                      QFE=xdaily - mean(xdaily, na.rm=T),
                      Station=x$Station[1],
                      stringsAsFactors=FALSE)
  } else {
    out <- data.frame(Time=NA, QFE=NA, Station=x$Station[1], stringsAsFactors=FALSE)
  }
  return(out)
}

daily <- lapply(PP, dailyfun)
dailymerge <- as.data.frame(Reduce(rbind, daily))
submerge <- dailymerge[dailymerge$Time == 'evening' & !is.na(dailymerge$QFE),]
submerge <- submerge[order(submerge$QFE),]
submerge$Station <- paste0('  ', gsub('_', ' ', submerge$Station)) ## for indent

ggplot(data=dailymerge[!is.na(dailymerge$Time), ], aes(Time, QFE)) +
  geom_line(aes(colour=Station, group=Station)) +
  scale_x_discrete(limits=c('morning', 'noon', 'evening', ''),
                   labels=c('morning (before 10AM)', 'noon (10AM-3PM)', 'evening (after 3PM)', '')) +
  guides(colour=FALSE) +
  theme(panel.grid.minor.x=element_blank(), panel.grid.major.x=element_blank()) +
  geom_text(data=submerge[c(1:3, 6, 18, nrow(submerge) - 2:0),],
            aes(label=Station, hjust=0)) +
  ylab('Station pressure anomalies from daily mean (hPa)')


## compute daily mean pressure and rename QFF to station name for later merging
PPdaily <- lapply(PP, function(x){
  xout <- aggregate(x$QFF, by=x[,c('Year', 'Month', 'Day')], mean, na.rm=T)
  names(xout)[4] <- x$Station[1]
  return(xout)
})

## Merge the data frames
## make sure we keep all the values
## (and not only rows that are present in all stations)
mergefun <- function(x,y) merge(x,y, all=TRUE)
dailymerge <- Reduce(mergefun, PPdaily)

## compute the correlation
ppcor <- cor(dailymerge[,-c(1:3)], use='p')
```

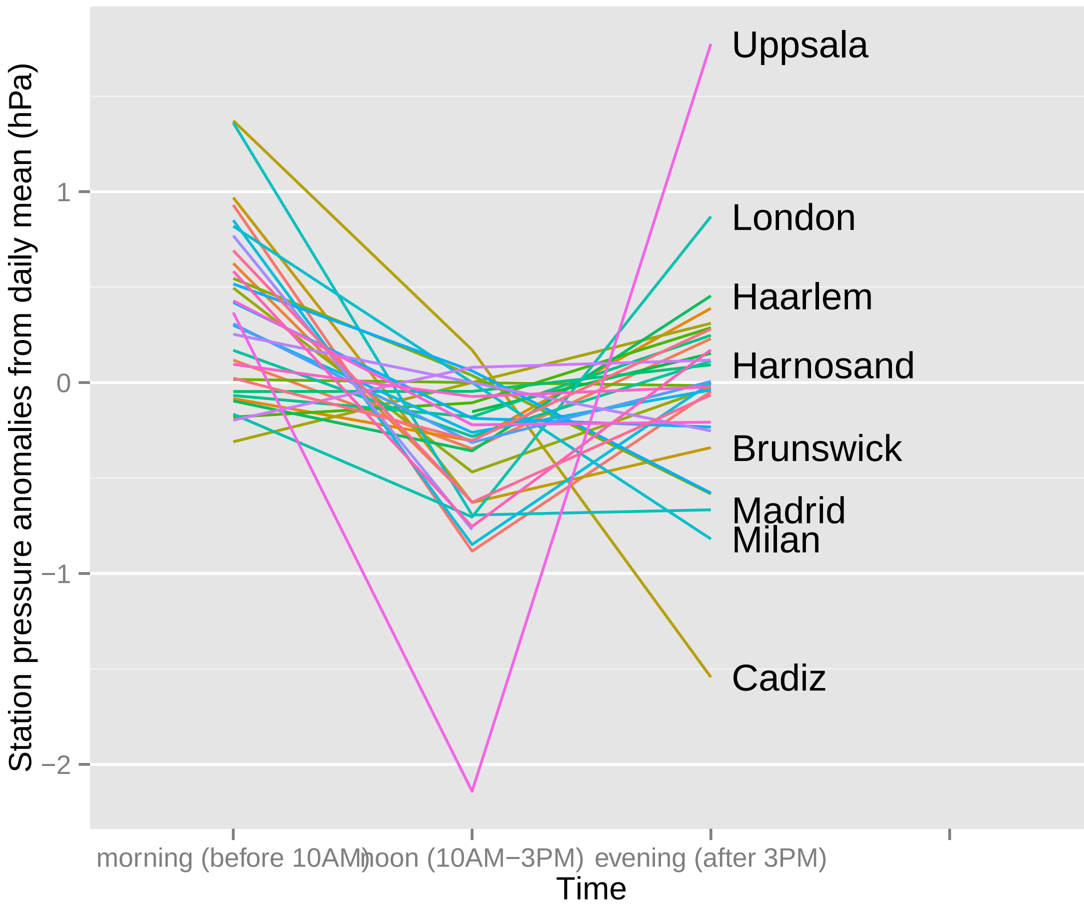

Figure 6: Daily cycle of station pressure (QFE, or QFF where QFE is not available)

```
## compute the distance between stations
lola <- sapply(PP[gsub(' ', '', rownames(ppcor))],
               function(x) unlist(x[1,c('Longitude', 'Latitude')]))
ppdist <- compute_dist(lola[1,], lola[2,], lola[1,], lola[2,])

df <- data.frame(Distance=ppdist[lower.tri(ppdist, diag=TRUE)],
                 Correlation=ppcor[lower.tri(ppcor, diag=TRUE)])

ggplot(data=df, aes(x=Distance, y=Correlation)) +
  geom_point() +
  geom_smooth(se=FALSE, method='loess', span=0.5) +
  xlab('Distance (in km)')
```

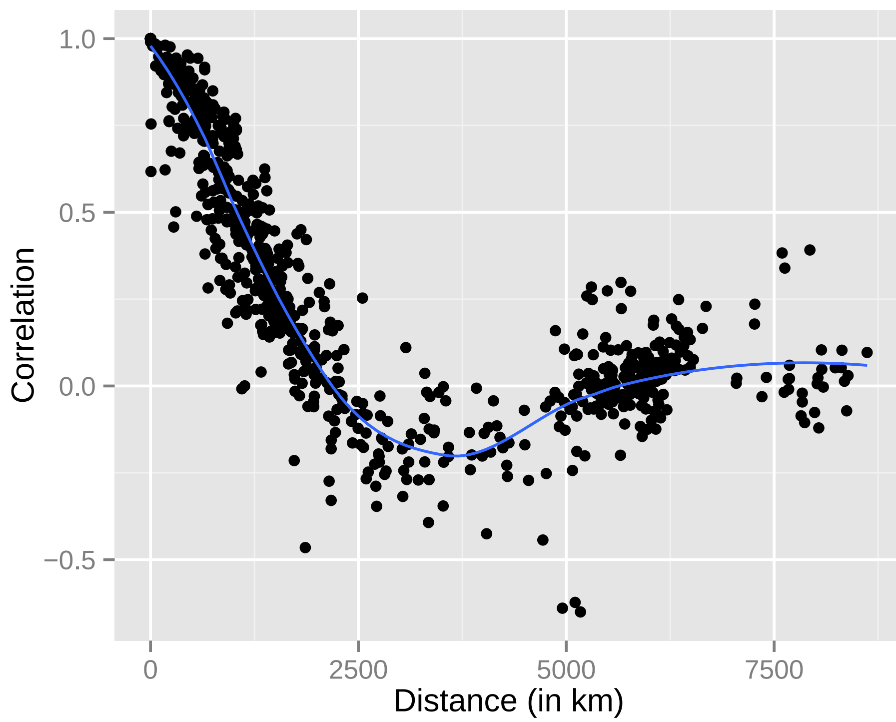

Figure 7: Correlation of daily mean sea level pressure at different stations by distance along with a smoothed fit (piecewise linear regression)

5 Analysis of temperature data at stations

```
ggplot(data=PPmerge[!is.na(PPmerge$TA.orig) & !is.na(PPmerge$TA.20CR),],
       aes(x=TA.20CR, y=TA.orig)) +
  geom_point() +
  geom_abline(c(0,1), colour='red') +
  geom_smooth(method='lm', se=FALSE) +
  facet_wrap(~ Station, ncol=5) +
  xlab('Climatological temperature from 20th century reanalysis') +
  ylab('Outside air temperature')
```

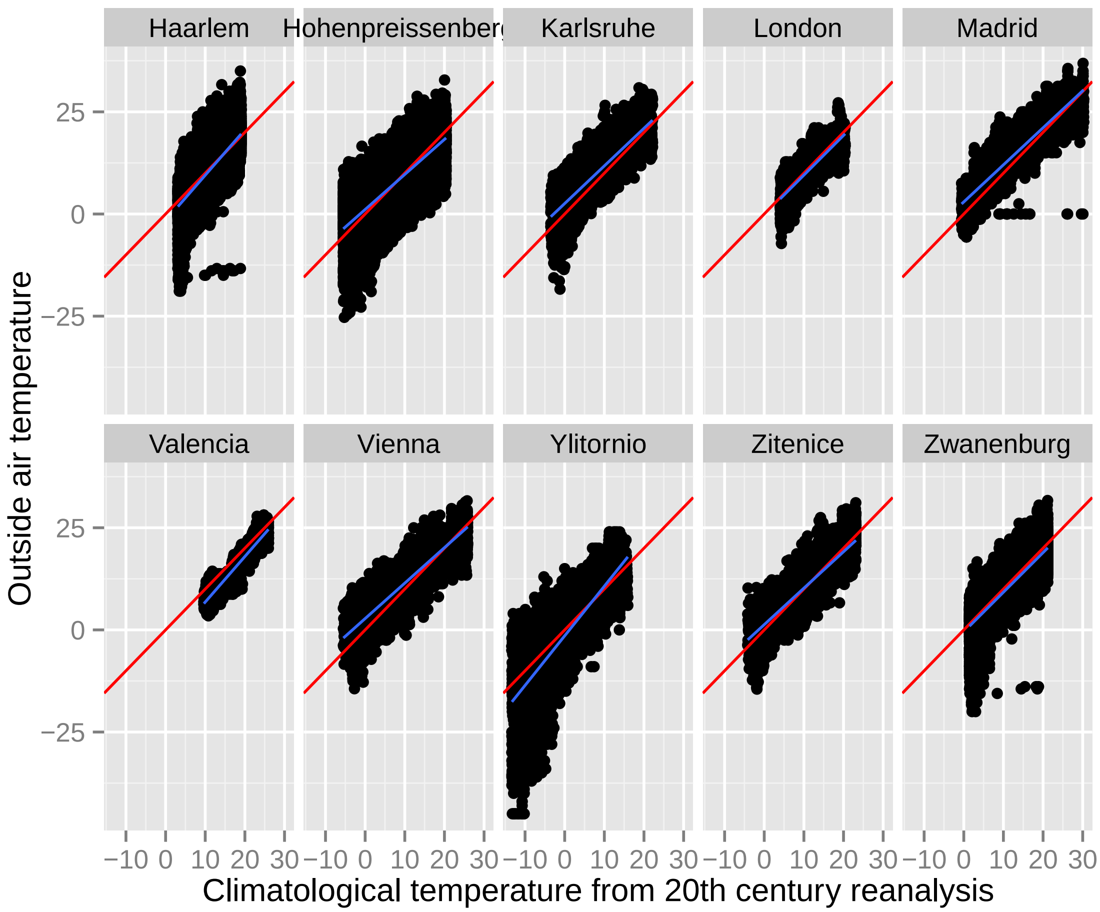

Figure 8: Observed in-situ air temperature vs. climatological temperature from 3-hourly temperature of the 20th century reanalysis (1871-1900) along with the 1-1 line in red and the best fit line in blue

```
ggplot(data=PPmerge[!is.na(PPmerge$TP.orig) & !is.na(PPmerge$TP.20CR),],
       aes(x=TP.20CR, y=TP.orig)) +
  geom_point() +
  geom_abline(c(0,1), colour='red') +
  geom_smooth(method='loess', se=FALSE) +
  facet_wrap(~ Station, ncol=4) +
  xlab('Climatological temperature from 20th century reanalysis') +
  ylab('Temperature at the barometer')
```

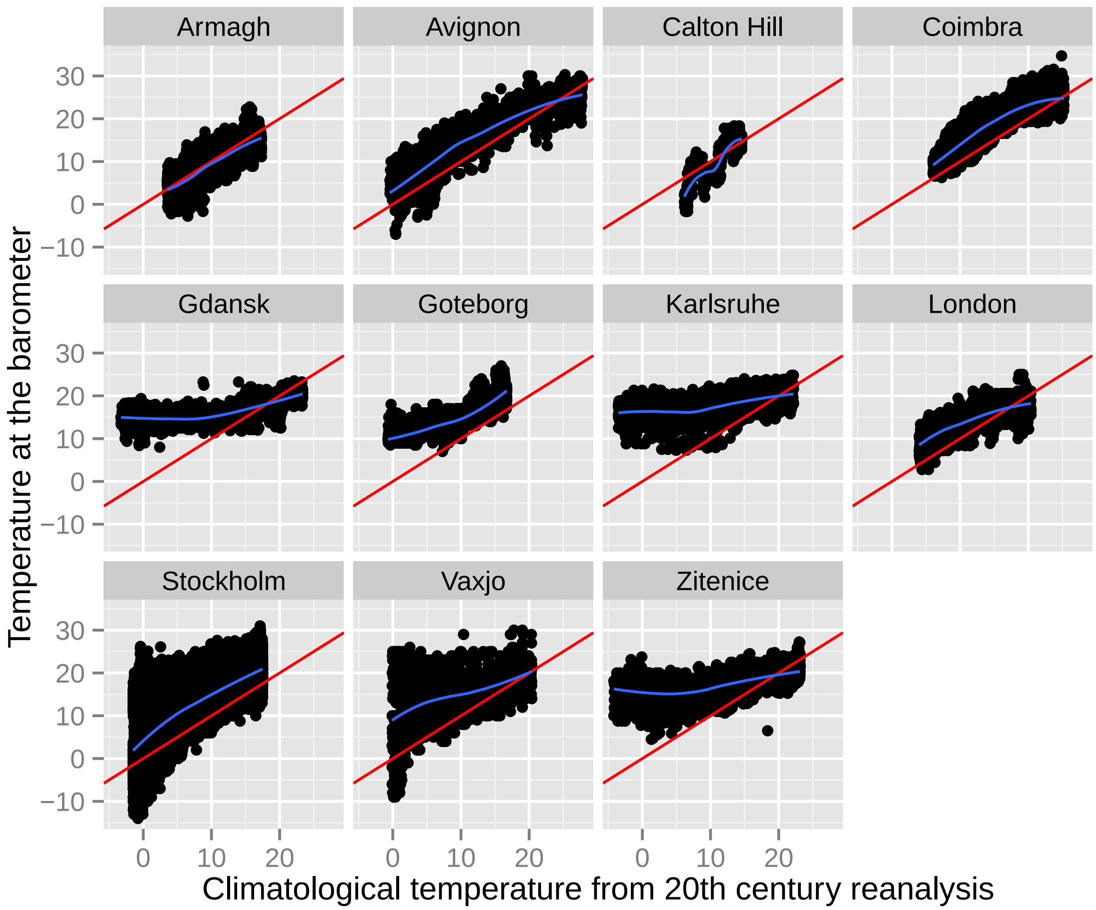

Figure 9: Observed temperature at the barometer vs. climatological temperature from 3-hourly temperature of the 20th century reanalysis (1871-1900) along with the 1-1 line in red and the best fit line in blue (using piecewise linear regression)

```
monnames <- format(as.Date('1990-01-15') + seq(0,330,30), '%b')

qplot(x=Month, y=TP.orig,
      group=Station, colour=Latitude,
      data=PPmerge[!is.na(PPmerge$TP.orig),],
      stat='summary', fun.y='mean', geom='line') +
  guides(colour=FALSE) +
  scale_x_discrete(name='Month', labels=monnames) +
  ylab('Temperature at barometer in \u00b0C')
```

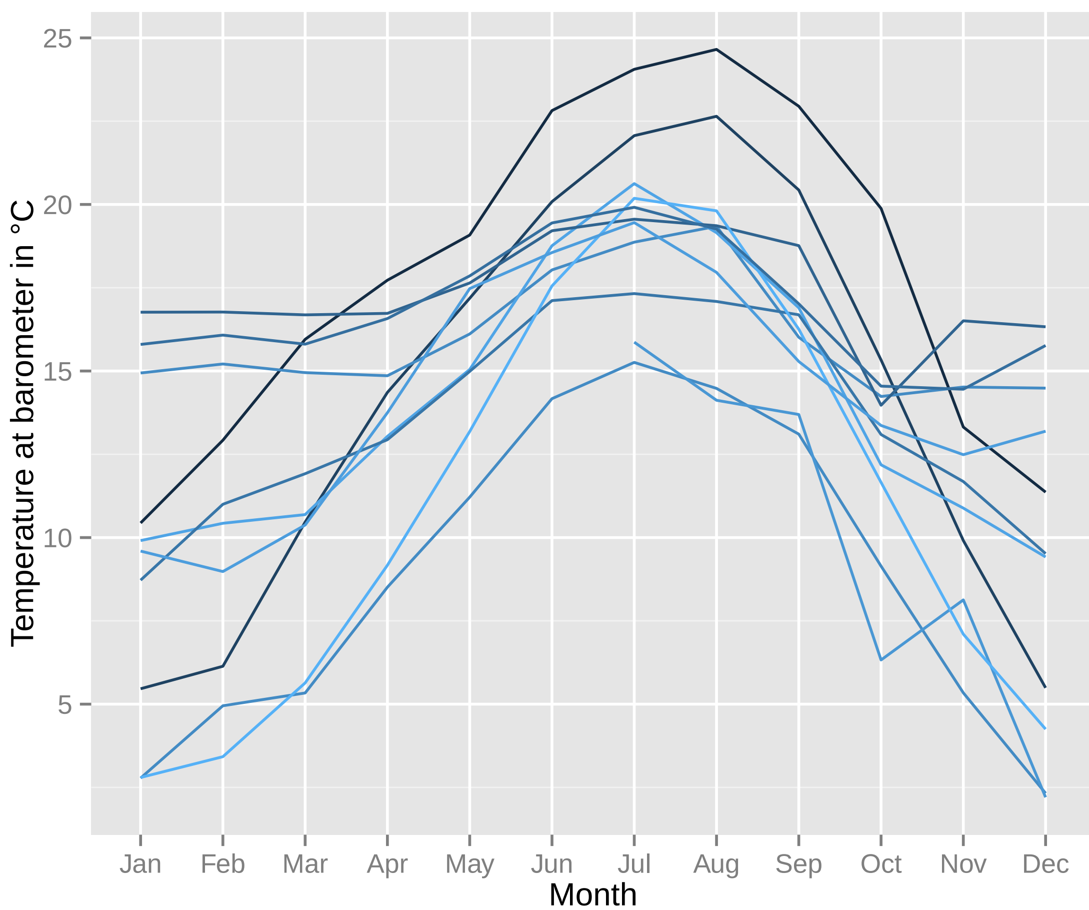

Figure 10: Monthly mean temperature at the barometer for stations with in-situ measurements. The line colour denotes the station latitude.

```
monnames <- format(as.Date('1990-01-15') + seq(0,330,30), '%b')

qplot(x=Month, y=TA.orig,
      group=Station, colour=Latitude,
      data=PPmerge[!is.na(PPmerge$TA.orig),],
      stat='summary', fun.y='mean', geom='line') +
  guides(colour=FALSE) +
  scale_x_discrete(name='Month', labels=monnames) +
  ylab('Outside air temperature in \u00b0C')
```

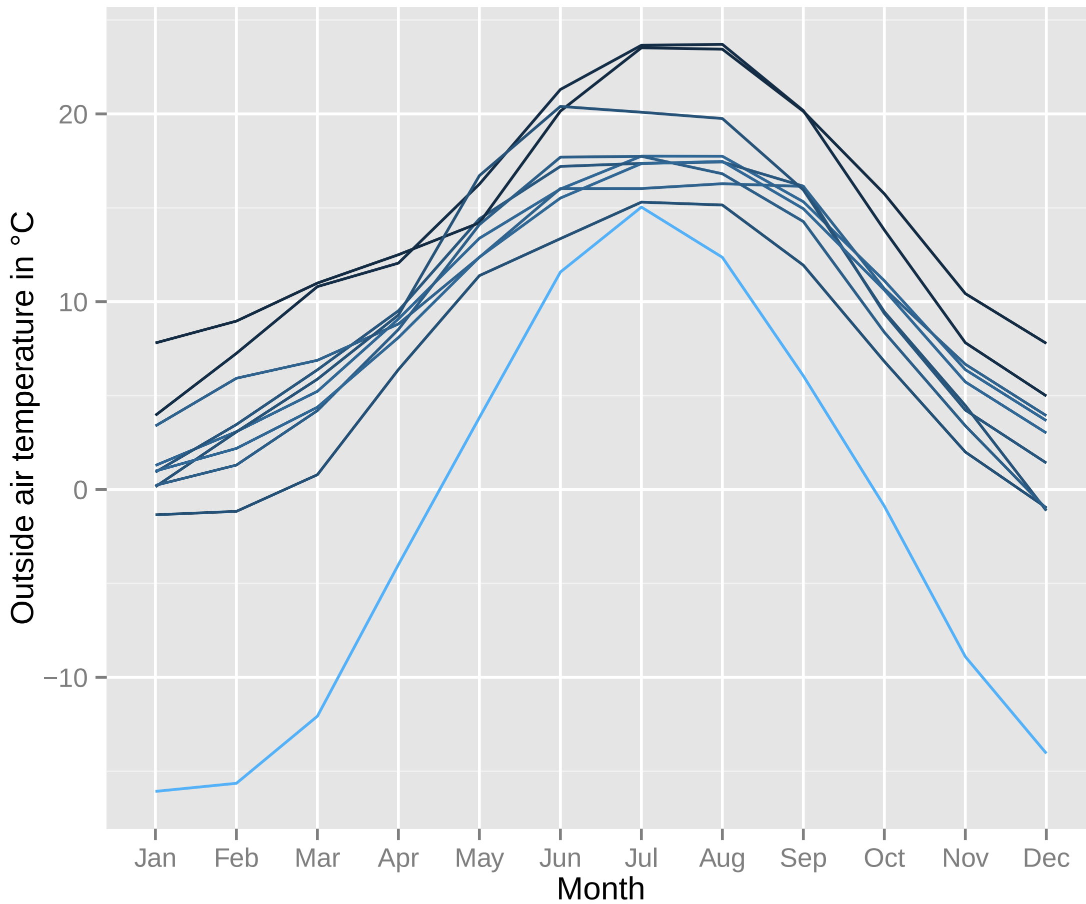

Figure 11: Monthly mean outside air temperature for stations with in-situ measurements. The line colour denotes the station latitude.

6 Analysis of travel documents

```
load('../data/all_station_and_travel_pressure_data.Rdata')
PPtravel <- PPmerge[PPmerge$Station %in% travelnames, ]

ggplot(data=PPtravel,
       aes(x=Local.date, y=QFE, group=Station, colour=Station)) +
  geom_line() + geom_point() +
  xlab('') + ylab('Absolute pressure at measurement location (hPa)')
```

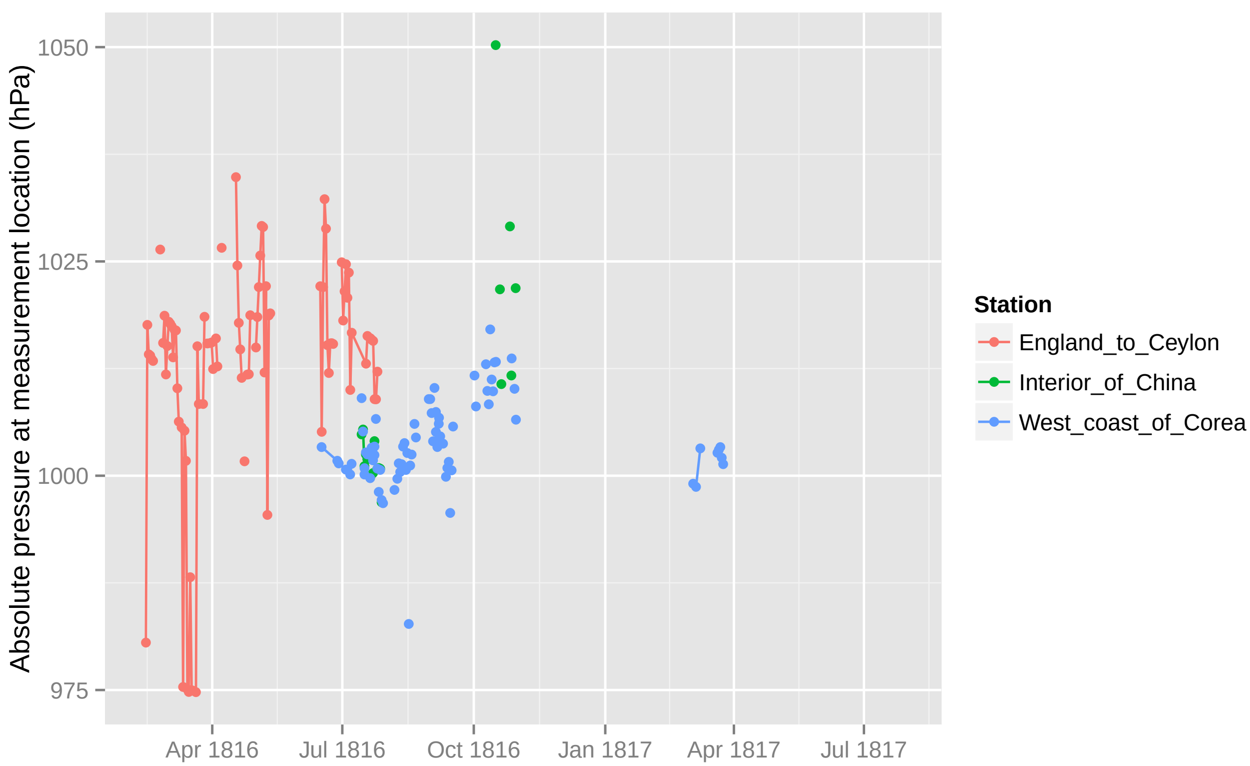

Figure 12: Time series of pressure records from travel documents

```
## compute bounding box
b <- t(sapply(PPtravel[,c('Longitude', 'Latitude')], range, na.rm=T))
b[1, ] <- (b[1, ] - mean(b[1, ])) * 1.05 + mean(b[1, ])
b[2, ] <- (b[2, ] - mean(b[2, ])) * 1.05 + mean(b[2, ])

## compute colour according to time
tlevs <- pretty(PPtravel$Local.date, 20)
tcols <- hcl(seq(0,300,length=length(tlevs) - 1), l=50, c=70)
PPtravel$colour <- cut(PPtravel$Local.date, tlevs)
## get station identifier (numeric) from 1-3
PPtravel$Stat <- as.numeric(factor(PPtravel$Station))
PPtravel$pch <- c(15,14,16)[PPtravel$Stat]
## get locations where we do not have pressure data
noQFF <- is.na(PPtravel$QFF)
PPtravel$pch[noQFF] <- c(1,0,2)[PPtravel$Stat[noQFF]]

map(xlim=b[1,], ylim=b[2,], interior=F)
box()
points(Latitude ~ Longitude, data=PPtravel, cex=0.8, lwd=2,
       pch=PPtravel$pch, col=tcols[PPtravel$colour])
legend('bottomright', gsub('_', ' ', unique(PPtravel$Station)),
       pch=14 + as.numeric(factor(unique(PPtravel$Station))),
       bg='white', inset=0.02, cex=0.8)
```

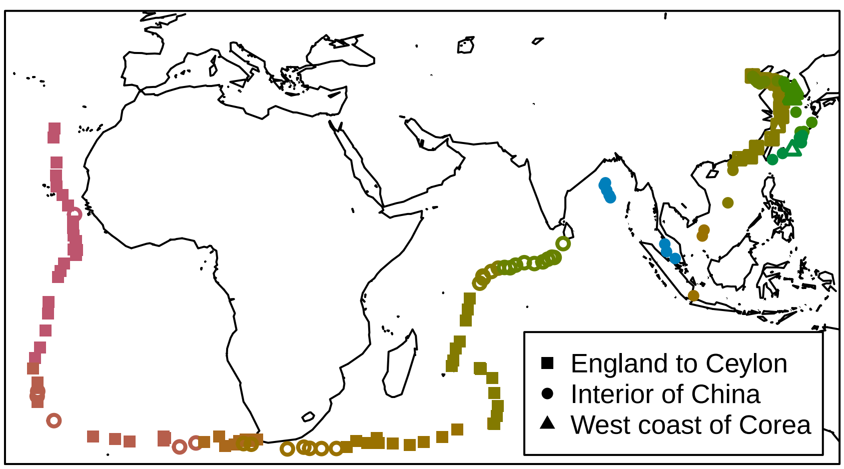

Figure 13: Trajectory of travel documents where quantitative information on observing location is known. Filled symbols denote locations for which pressure data are available, colours indicate time from February 1816 (red) to July 1817 (blue).

7 Appendix 1: Overview Table

	Period	Number of records	Number of records in 1815-17
Albany	1815-1815	542	542
Armagh	1815-1817	3286	3286
Avignon	1816-1816	982	982
Barcelona	1811-1820	10943	3288
Barnton	1815-1817	968	968
Boston	1816-1817	711	711
Brunswick	1815-1817	3112	3112
Cadiz	1816-1820	4749	1461
CaltonHill	1817-1817	340	340
Cambridge	1815-1816	818	818
Coimbra	1815-1817	3665	3665
Derby	1817-1817	64	64
England_to_Ceylon	1816-1816	86	86
Gdansk	1815-1817	3278	3278
Geneva	1796-1820	9026	1065
Goteborg	1815-1817	2192	2192
Haarlem	1801-1840	43830	3288
Harnosand	1815-1816	2027	2027
Hohenpreissenberg	1781-1820	37203	3288
Interior_of_China	1816-1816	21	21
Karlsruhe	1815-1817	3288	3288
London	1815-1817	2192	2192
Madrid	1814-1817	2580	1488
Milan	1778-1834	41499	2190
Natchez	1815-1817	2210	2210
NewBedford	1815-1817	4384	4384
NewHaven	1815-1817	789	789
Nuuk	1816-1820	4846	2102
Paris_a	1811-1820	3649	1092
Paris_b	1816-1817	2920	2920
Prague	1815-1817	1096	1096
QuebecCity	1798-1819	10812	2183
Salem	1786-1820	24672	2145
Stockholm	1756-1858	101913	3286
Turin	1787-1865	26844	1096

	Period	Number of records	Number of records in 1815-17
Umea	1815-1817	1096	1096
Uppsala	1760-1838	55585	2194
Valencia	1815-1817	2309	2309
Vaxjo	1815-1817	3288	3288
Vienna	1815-1817	3246	3246
West_coast_of_Corea	1816-1817	113	113
Ylitornio	1800-1822	7361	1090
Zitenice	1815-1818	4383	3288
Zwanenburg	1801-1860	57974	3288