Long term station data analysis

Brian Yurk

12/4/2020

```
rm(list=ls())
gc()
```

```
##          used (Mb) gc trigger (Mb) limit (Mb) max used (Mb)
## Ncells 423889 22.7     882072 47.2         NA   658077 35.2
## Vcells 808217  6.2    8388608 64.0      32768  1802945 13.8
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(ggplot2)
```

Some functions here. A few only do one thing.

```
compute_durations <- function(tib){ #computes time elapsed between measurements (hours)
  tib <- tib %>% mutate(dt = as.duration(lag(datetime,1) %--% datetime)/dhours(1))
  return(tib)
}

#computes drift potential from drift potential per time and time interval width
compute_dpt <- function(tib){
  tib <- tib %>% mutate(dpt = dp*dt)
  return(tib)
}

#computes modified drift potentil by not allowing time intervals to exceed 3 hours.
#For those that do, the remainder of the time interval is recorded
compute_dpt2 <- function(tib,dtmax=3){
  tib <- tib %>% mutate(dpt2 = dp*pmin(dt,3),lost_dt = dt-pmin(dt,3))
  return(tib)
}

#applies the previous functions after dropping na wind speeds
compute_dp_stuff <- function(tib,dtmax = 3){
  tib <- tib %>% drop_na(WS) %>% compute_durations() %>%
    compute_dpt() %>% compute_dpt2(dtmax)
  return(tib)
}

drop_yrs <- function(tib,yrs){ #drops specified years from data
  tib <- tib[!(year(tib$datetime) %in% yrs),]
  return(tib)
}

#yearly totals: drift potential, by direction, by coast;
#lost time, number of observations, etc
compute_by_yr_stuff_directional <- function(tib){
  tib <- tib %>% select(datetime,dt,dp,WD,dpt,dpt2,lost_dt) %>%
    mutate(yr = year(datetime)) %>% group_by(yr) %>%
    summarize(dp_m = mean(dp,na.rm=TRUE),dt_m=mean(dt,na.rm=TRUE),dt_sd=sd(dt,na.rm=TRUE),
              dpt_su=sum(dpt,na.rm=TRUE),dpt2_su=sum(dpt2,na.rm=TRUE),
              lost_dt_su=sum(lost_dt,na.rm=TRUE),
              dp_n = sum(dpt2*(WD>=0)*(WD<11.25)+dpt2*(WD>=348.75)*(WD<=360),na.rm=TRUE),
              dp_nne = sum(dpt2*(WD>=11.25)*(WD<33.75),na.rm=TRUE),
              dp_ne = sum(dpt2*(WD>=33.75)*(WD<56.25),na.rm=TRUE),
              dp_ene = sum(dpt2*(WD>=56.25)*(WD<78.75),na.rm=TRUE),
              dp_e = sum(dpt2*(WD>=78.75)*(WD<101.25),na.rm=TRUE),
              dp_ese = sum(dpt2*(WD>=101.25)*(WD<123.75),na.rm=TRUE),
              dp_se = sum(dpt2*(WD>=123.75)*(WD<146.25),na.rm=TRUE),
              dp_sse = sum(dpt2*(WD>=146.25)*(WD<168.75),na.rm=TRUE),
              dp_s = sum(dpt2*(WD>=168.75)*(WD<191.25),na.rm=TRUE),
              dp_ssw = sum(dpt2*(WD>=191.25)*(WD<213.75),na.rm=TRUE),
              dp_sw = sum(dpt2*(WD>=213.75)*(WD<236.25),na.rm=TRUE),
              dp_wsw = sum(dpt2*(WD>=236.25)*(WD<258.75),na.rm=TRUE),
              dp_w = sum(dpt2*(WD>=258.75)*(WD<281.25),na.rm=TRUE),
              dp_wnw = sum(dpt2*(WD>=281.25)*(WD<303.75),na.rm=TRUE),
              dp_nw = sum(dpt2*(WD>=303.75)*(WD<326.25),na.rm=TRUE),
              dp_nnw = sum(dpt2*(WD>=326.25)*(WD<348.75),na.rm=TRUE),
              dp_sc = sum(dpt2*(WD>=0)*(WD<=45)+dpt2*(WD>=315)*(WD<=360),na.rm=TRUE),
              dp_swc = sum(dpt2*(WD>=0)*(WD<=90)+dpt2*(WD==360),na.rm=TRUE),
              dp_wc = sum(dpt2*(WD>=45)*(WD<=135),na.rm=TRUE),
              dp_nwc = sum(dpt2*(WD>=90)*(WD<=180),na.rm=TRUE),
              dp_nc = sum(dpt2*(WD>=135)*(WD<=225),na.rm=TRUE),
              dp_nec = sum(dpt2*(WD>=180)*(WD<=270),na.rm=TRUE),
              dp_ec = sum(dpt2*(WD>=225)*(WD<=315),na.rm=TRUE),
              dp_sec = sum(dpt2*(WD>=270)*(WD<=360)+dpt2*(WD==0),na.rm=TRUE),
              nobs=n())
  return(tib)
}

#drop years with more than 175 (default) hours of lost time
```

```
drop_big_lost_dt_yrs<- function(tib,dtmax=175){
  tib <- tib[tib$lost_dt_su<=dtmax,]
  return(tib)
}

#a function to group the data by years - used for decadal computations
group_by_yrs <- function(tib,breaks){
  tib <- tib %>%
    mutate(num_yrs = 1) %>%
    group_by(gr=cut(yr,breaks=breaks,right=FALSE)) %>%
    summarize_at(vars(dpt2_su:num_yrs),sum,na.rm=TRUE)
  return(tib)
}

dp_plot <- function(dp,yscale=30){ #plot fryberger diagrams
  dir <- seq(0,360-22.5,by=22.5)
  ndp <- dp*cos(dir*pi/180) #north components of dp
  edp <- dp*sin(dir*pi/180) #east components of dp
  rndp <- sum(ndp) #north component of resultant dp
  redp <- sum(edp) #east component of resultant dp
  rdp <- sqrt(rndp^2+redp^2) #rdp magnitude
  #rdp direction (rotated 180 to point where the sand is going)
  rdir <- (atan2(redp,rndp)*180/pi+180) %% 360

  dp_df <- structure(list(dir = dir, dp = dp),.Names=c("dir", "dp"),
                     class="data.frame",row.names=c(NA,-16) )

  rdp_df <- structure(list(dir = rdir, dp = rdp),.Names=c("dir", "dp"),
                      class="data.frame",row.names=c(17L) )

  #normalize by RDP - details are weird. Set to match scaling of an earlier plot
  if(yscale=="RDP"){yscale <- 2.7e7*rdp_df[1,2]/13075309}

  #Set ylim to be as large as needed for largest dp in set of diagrams being
  # displayed together. Then we get consistent scaling
  base <- ggplot(dp_df, aes(x=dir,y=dp))
  p <- base + coord_polar() + ylim(0,yscale) +
    scale_x_continuous(limits=c(0,360),breaks=dir)
  q <- p + geom_segment(data = dp_df , aes(y=0,xend=dir,yend=dp),col="black") +
    geom_segment(data = rdp_df , aes(y=0,xend=dir,yend=dp),
                 arrow=arrow(length=unit(0.3,"cm")),col="grey",size=1) +
    theme(axis.line=element_blank(),axis.text.x=element_blank(),
          axis.text.y=element_blank(),axis.ticks=element_blank(),
          axis.title.x=element_blank(),
          axis.title.y=element_blank(),legend.position="none",
          panel.background=element_blank(),panel.border=element_blank(),
          panel.grid.major=element_blank(),
          panel.grid.minor=element_blank(),plot.background=element_blank())
  #print(q)
  return(q)
}

#plot and maybe save fryberger diagram for a particular row of the data
```

```r
dp_plot_yrs <- function(tib,row,yscale=30,fname=NULL){
  st <- which(names(tib)=="dp_n")
  en <- which(names(tib)=="dp_nnw")
  dp <- as.numeric(tib[row,st:en])
  #print(dp)
  #print(str(dp))
  dp_plot(dp,yscale=yscale)
  if(!is.null(fname)){
    ggsave(fname,height = 3, width=3, units= "in")
  }
}

#Compute directional dps as proportion of total dp
compute_rel_dp <- function(tib){
  levs <- rev(c("e","ese","se","sse","s","ssw","sw","wsw","w","wnw",
              "nw","nnw","n","nne","ne","ene"))
  tib <- tib %>% mutate_at(vars(dp_n:dp_nnw),funs(./dpt2_su)) %>%
    select(station:dp_nnw) %>% select(-(dp_m:lost_dt_su)) %>%
    pivot_longer(!c(station,yr),names_to="dir",values_to="dp") %>%
    mutate(dir=factor(gsub(".*_","",dir),levels=levs))
  return(tib)
}

#regression analyses for dp vs yr
lm_dp_vs_yr <- function(tib,start_yr=1961,with_plots=TRUE){
  lm_dp_yr <- tib %>% select(yr,dpt2_su) %>% filter(yr>=start_yr) %>%
    lm(dpt2_su ~ yr, data = .) %>% summary()
  if(with_plots){
    tib <- tib %>% select(yr,dpt2_su) %>% filter(yr>=start_yr)
    p <- ggplot(data=tib, mapping=aes(x=yr,y=dpt2_su)) + geom_point() +
      ylab('drift potential') + xlab('year') +
      geom_smooth(method='lm')
    print(p)
  }
  return(lm_dp_yr)
}

#proportion change 1961-2019 based on linear model
lm_pred_prop_change <- function(lmsummary){
  m <- lmsummary$coefficients[2,1] #slope
  b <- lmsummary$coefficients[1,1] #intercept
  prop_change <- m*(2019-1961)/(b+m*1961)
  return(prop_change)
}
```

Import dp data for each station, drop rows with NA wind speeds, compute time intervals, compute drift potentials (q*dt, calling it dpt, dp is drift potential per time). Drop partial years at the beginning and end of the data. 1988 is missing from the data at BEH. Make a list of data frames to streamline what would otherwise be repetitive function calls.

```r
beh <- readRDS('clean_data/BEH_19730101_20201207_dp.rds') %>% compute_dp_stuff() %>%
  drop_yrs(c(1973,1988,2020))
biv <- readRDS('clean_data/KBIV_19961231_20201203_dp.rds') %>% compute_dp_stuff() %>%
  drop_yrs(c(1996,2020))
```

```
grb <- readRDS('clean_data/GRB_19490901_20201207_dp.rds') %>% compute_dp_stuff() %>%
  drop_yrs(c(1949,2020))
mdw <- readRDS('clean_data/MDW_19480101_20201207_dp.rds') %>% compute_dp_stuff() %>%
  drop_yrs(c(2020))
mke <- readRDS('clean_data/MKE_19471231_20201207_dp.rds') %>% compute_dp_stuff() %>%
  drop_yrs(c(1947,2020))
mkg <- readRDS('clean_data/KMKG_19480101_20201204_dp.rds') %>% compute_dp_stuff() %>%
  drop_yrs(c(2020))
tvc <- readRDS('clean_data/TVC_19481201_20201207_dp.rds') %>% compute_dp_stuff() %>%
  drop_yrs(c(1948,2020))

station_list <- list(beh=beh,biv=biv,grb=grb,mdw=mdw,mke=mke,mkg=mkg,tvc=tvc)
```

Compute yearly drift potentials, including by direction. The dpt2 variable computes drift potentials for at most a 3 hour duration. If the gap between reports is larger, then dpt2 is dp*3 (just the first 3 hours), and the rest of the duration is ignored (0 dp during the rest). The lost_dt variable measures how much time is lost. We remove years from the data where this adds up to more than 175 hours (2% of the year). In some cases, stations seemed to stop reporting at night (see, e.g., early BEH data). These years will not be represented in the data. In other cases there were large gaps in the data. These years are also removed.

```
lost_dt_max <- 175 #hrs
station_list_yr <- station_list %>% map(compute_by_yr_stuff_directional) %>%
  map(drop_big_lost_dt_yrs,dtmax=lost_dt_max) #as a list of tibbles
#as a single tibble (better for plotting)
station_yr <- station_list_yr %>% bind_rows(.id='station')
```

Now create decadal summaries. When doing decadal summaries this way, it is important to be mindful of missing years introduced in the previous step. This is especially important if looking at decadal dp totals, for example. To make comparisons of amounts of drift potential fair this should be divided by the number of years retained in the decade. The number of years retained is located in the num_yrs column. We do not do this sort of comparison. Decades are 1948-1957, 1958-1967,. . . , 2008-2017.

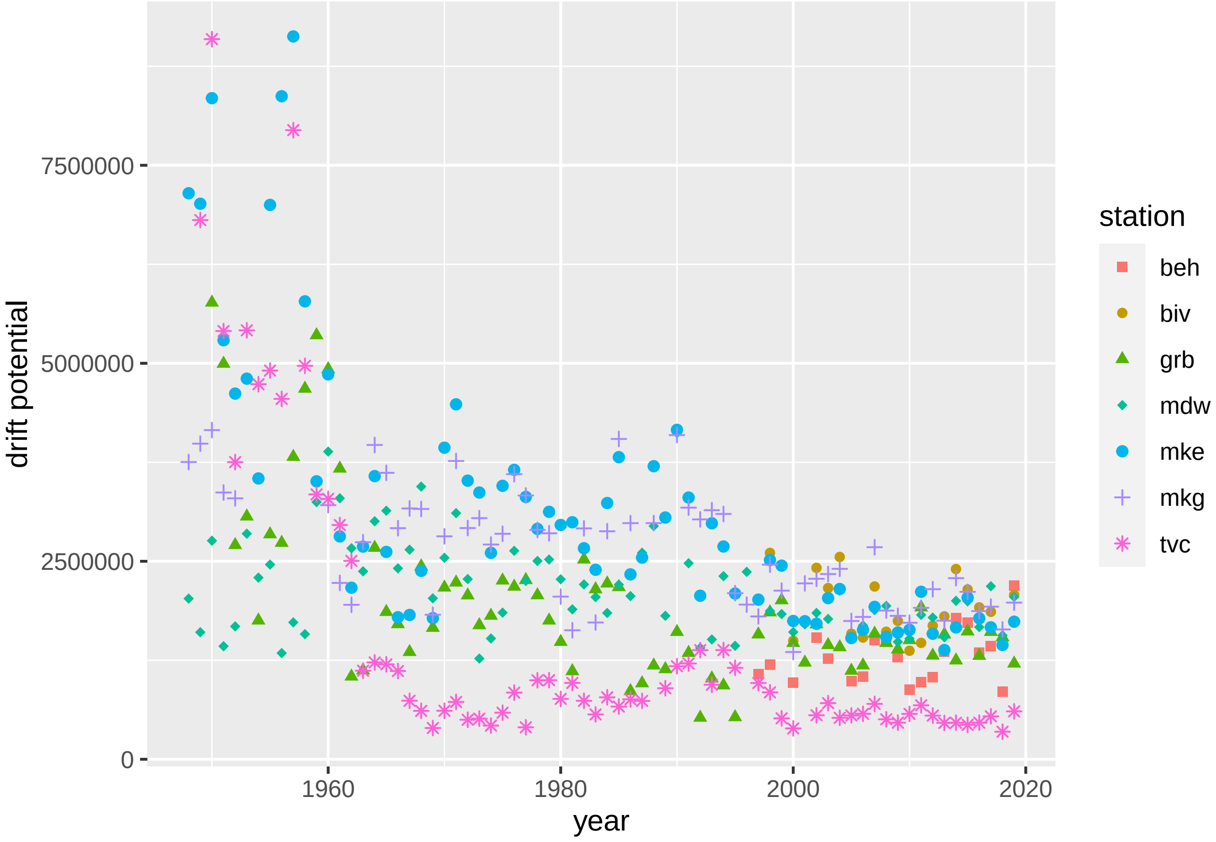
```
breaks <- seq(1948,2018,by=10)
#as a list of tibbles
station_list_dec <- station_list_yr %>% map(group_by_yrs,breaks=breaks)
station_dec <- station_list_dec %>% bind_rows(.id='station') #as a single tibble

stat_dec_numyrs <- station_dec %>% select(station,gr,num_yrs)
print(stat_dec_numyrs)
```

```
## # A tibble: 47 x 3
##    station gr                  num_yrs
##    <chr>   <fct>                 <dbl>
##  1 beh     [1.99e+03,2e+03)          1
##  2 beh     [2e+03,2.01e+03)          7
##  3 beh     [2.01e+03,2.02e+03)      10
##  4 beh     <NA>                      2
##  5 biv     [2e+03,2.01e+03)          8
##  6 biv     [2.01e+03,2.02e+03)      10
##  7 biv     <NA>                      2
##  8 grb     [1.95e+03,1.96e+03)       8
##  9 grb     [1.96e+03,1.97e+03)      10
## 10 grb     [1.97e+03,1.98e+03)      10
## # ... with 37 more rows
```

Plot the total drift potential for each year and for each station.

```
(p <- ggplot(data = station_yr,
             aes(x = yr, y = dpt2_su, shape = station, color = station)) +
   geom_point() +
   scale_shape_manual(values=c(15,16,17,18,19,3,8)) +
   ylab('drift potential') +
   xlab('year'))
```



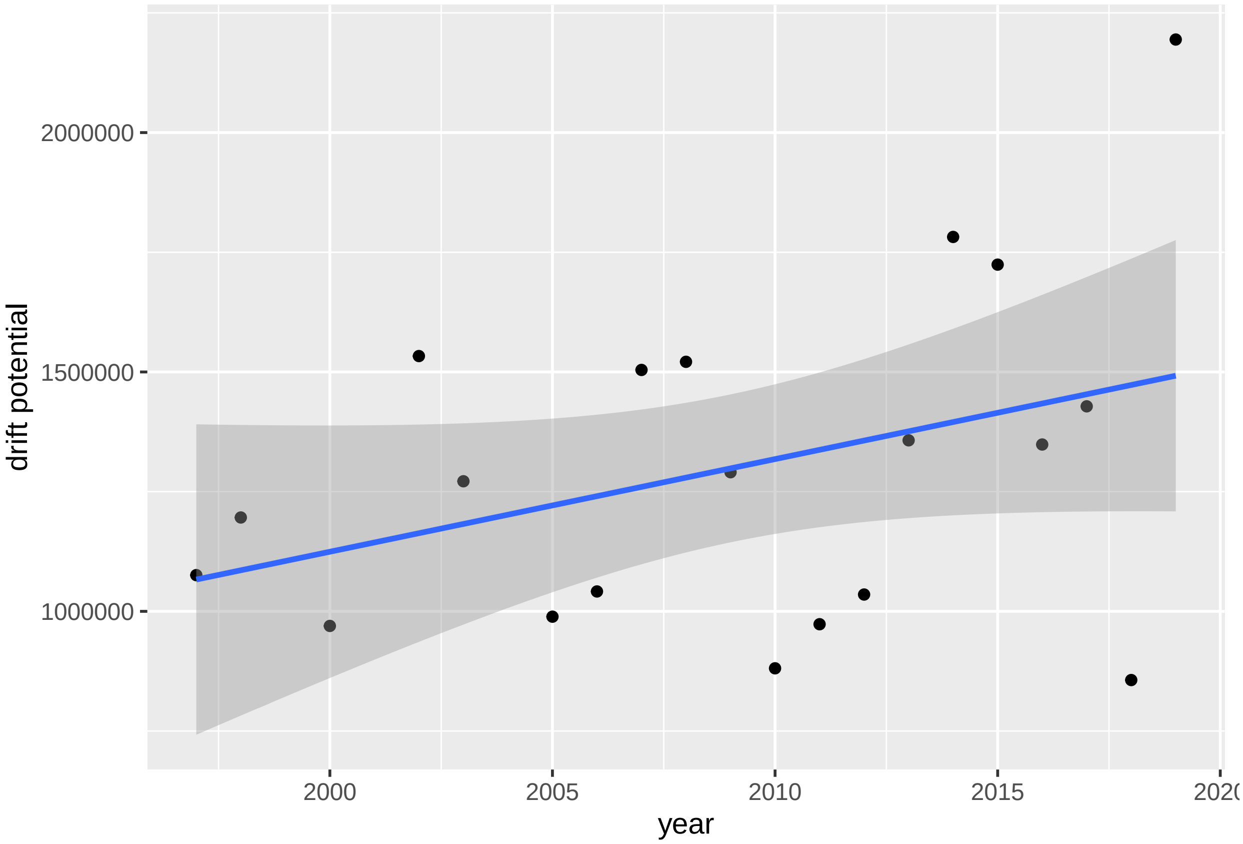
```
ggsave('outputs/dp_yearly.pdf')
```

```
## Saving 6.5 x 4.5 in image
```

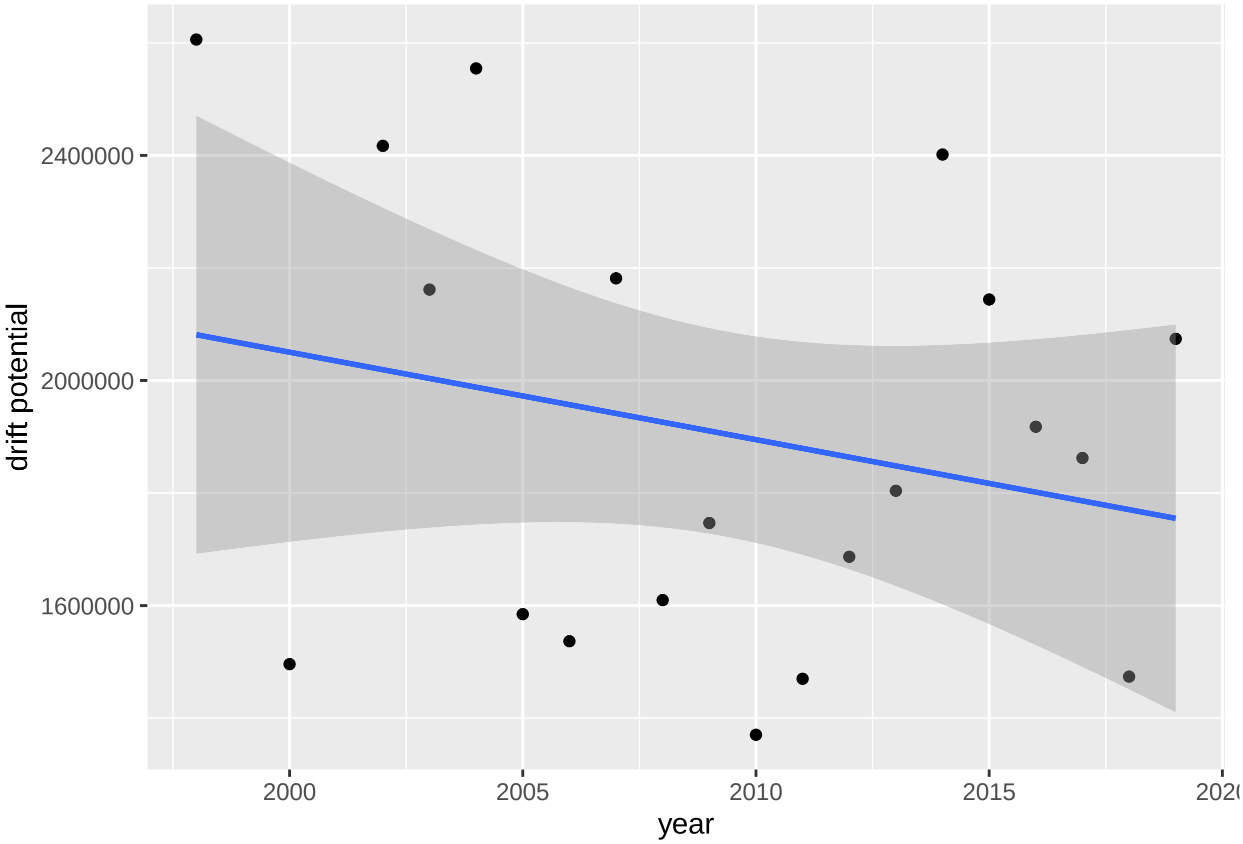
Regression analysis for each station, using only years after 1960.

```
lm_list <- station_list_yr %>% map(lm_dp_vs_yr,start_yr=1961,with_plots=TRUE)
```

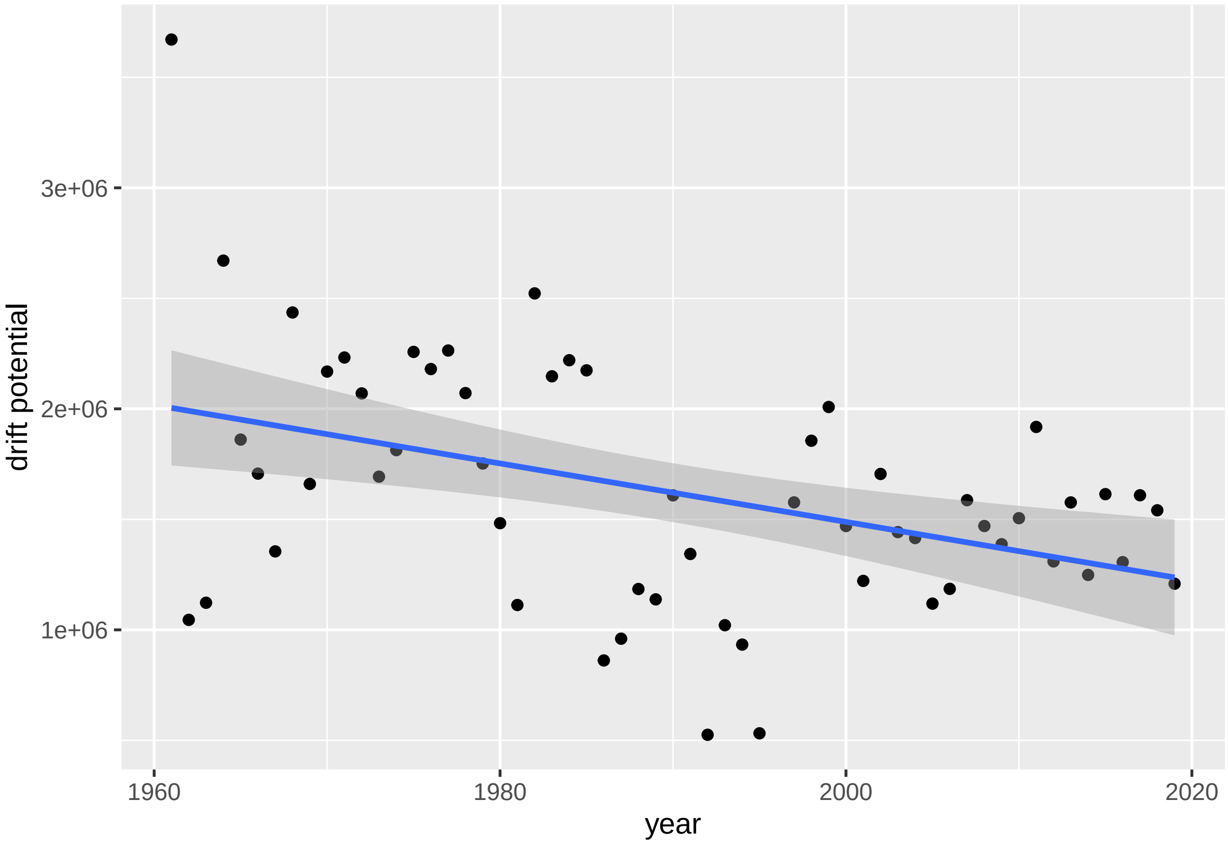
```
## `geom_smooth()` using formula 'y ~ x'
```



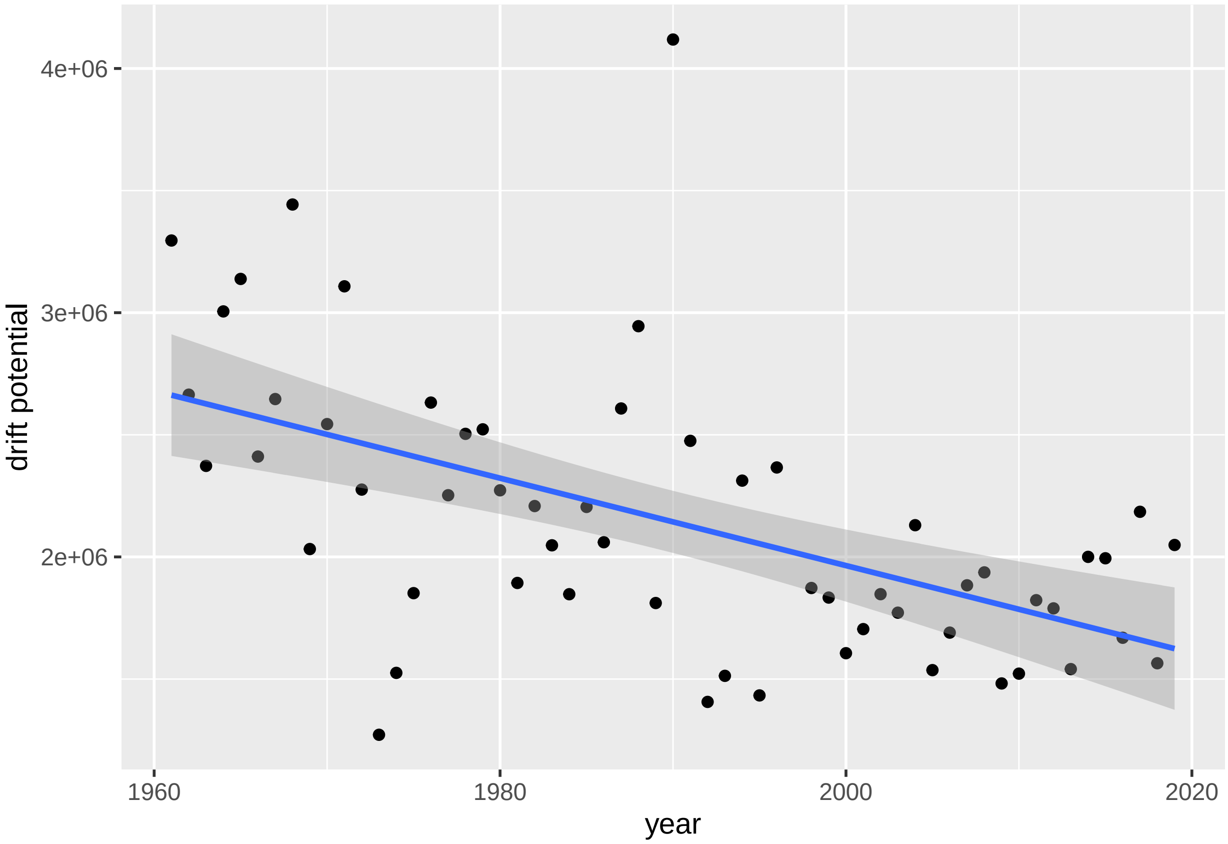
```
## `geom_smooth()` using formula 'y ~ x'
```



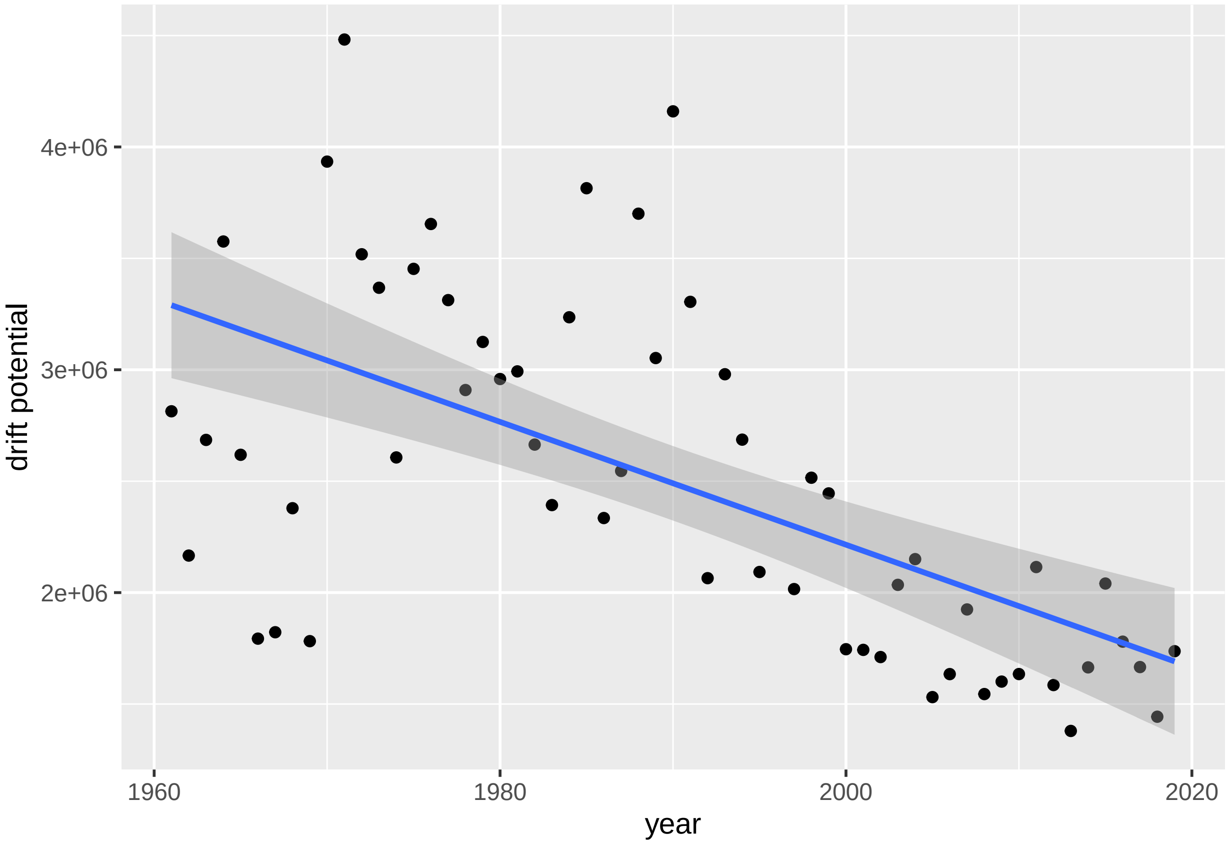
```
## `geom_smooth()` using formula 'y ~ x'
```



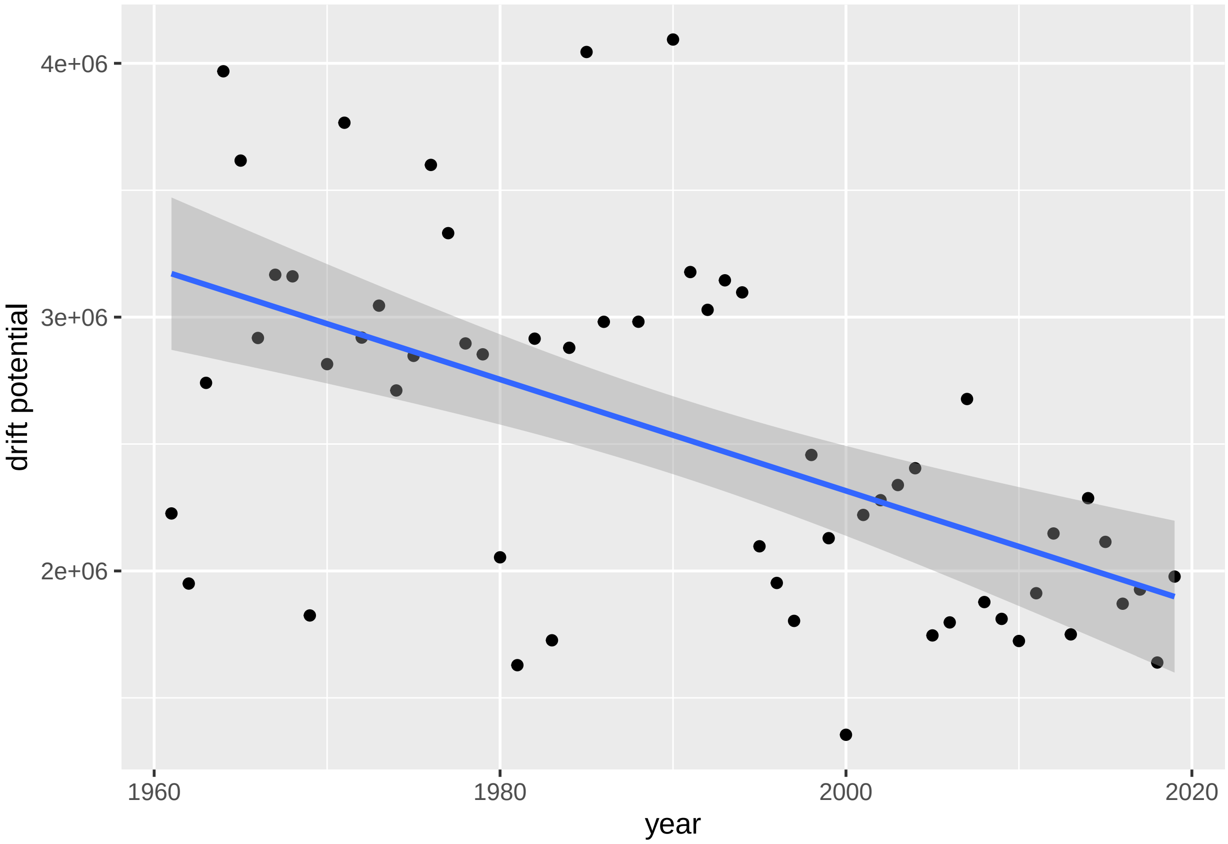
```
## `geom_smooth()` using formula 'y ~ x'
```



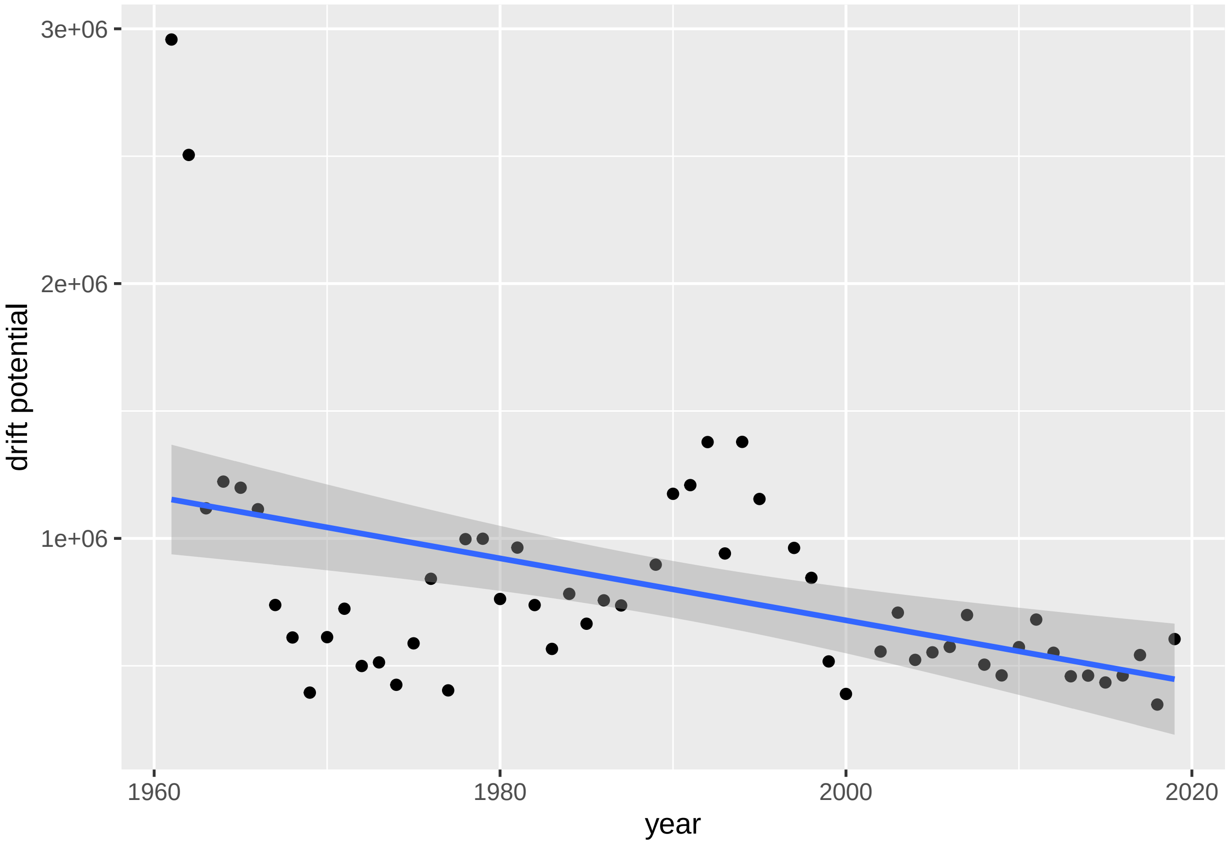
```
## `geom_smooth()` using formula 'y ~ x'
```



```
## `geom_smooth()` using formula 'y ~ x'
```



```
## `geom_smooth()` using formula 'y ~ x'
```



```
#proportion change over 1961-2019 using linear model
prop_change_list <- lm_list %>% map(lm_pred_prop_change)
```

Plot and save decadal Fryberger diagrams for 5 stations.

```
fry_table_stations <- c("mkg","tvc","mke","grb","mdw")
#yscales <- rep(2.7e7,5)
yscales <- rep("RDP",5) #normalize to rdp
ros <- 1:7
yr_int_str <- c("1948_1957","1958_1967","1968_1977","1978_1987","1988_1997","1998_2007",
                "2008_2017")


for(i in seq_along(fry_table_stations)){ #loop over stations
  for(j in seq_along(ros)){ #loop over decades
    #plots are saved as pdfs. These are the file names.
    fname <- paste0("outputs/",fry_table_stations[i],"_",yr_int_str[j],".pdf")
    dp_plot_yrs(station_list_dec[[fry_table_stations[i]]],ros[j],yscale=yscales[i],
                fname=fname)
  }
}
```

Compute relative drift potentials from each direction (16 directions) and plot stacked bar plots showing changes in drift potential direction over time for a single station.

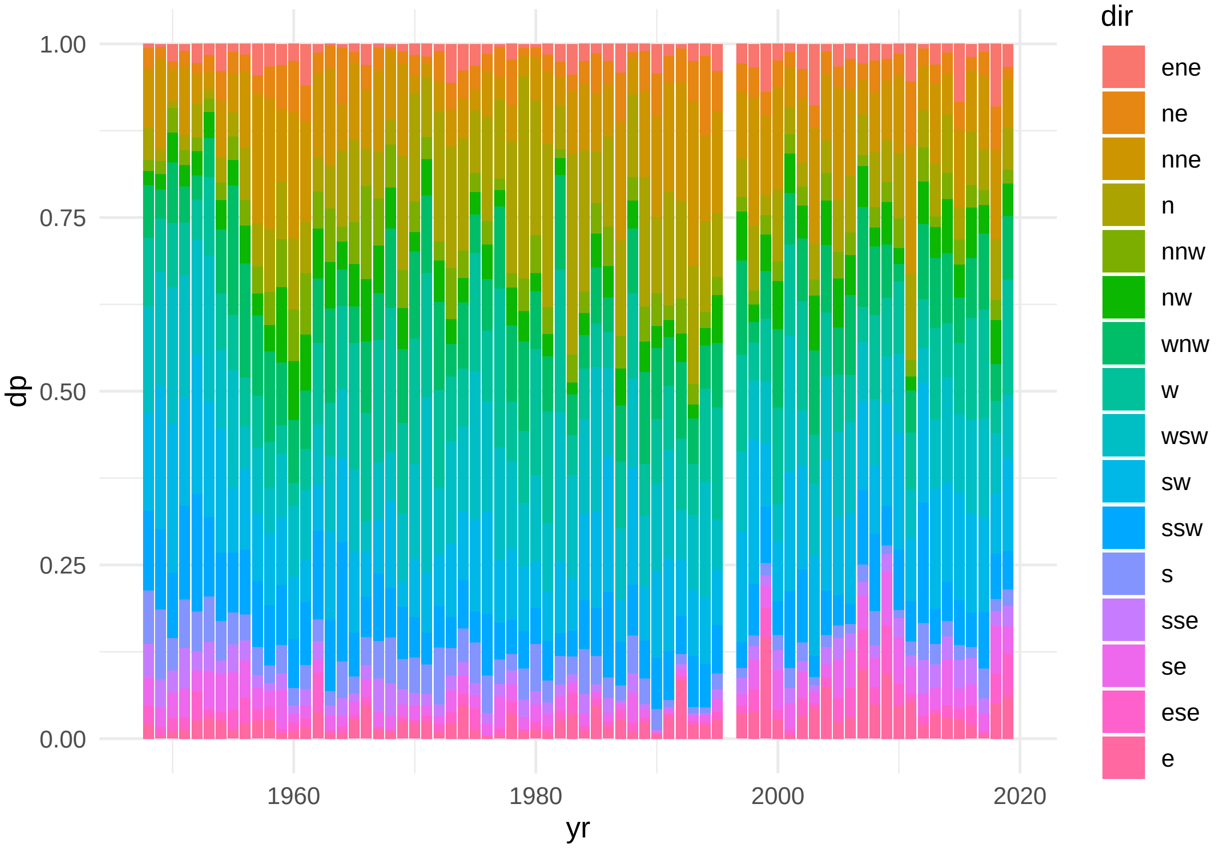
```
station_yr_rel <- station_yr %>% compute_rel_dp() #already in long format
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
```

```
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```
mke_rel <- station_yr_rel %>% filter(station=="mke")

(q <- ggplot(data = mke_rel, aes(x = yr, y = dp, fill = dir)) +
    geom_bar(stat="identity") +
    theme_minimal())
```



```
ggsave("outputs/mke_yr_rel_stack.pdf")
```

```
## Saving 6.5 x 4.5 in image
```