Metropolis Campaign

## 1 *Metropolis algorithm simulation*

*based on* [*Markov Chain Monte Carlo | Columbia Public Health*](https://www.publichealth.columbia.edu/research/population-health-methods/markov-chain-monte-carlo) *with slightly changed assumptions (10 counties, pre-defined populations either systematically chosen or randomized) and these rules:*

1. *Flip a coin. Heads to move east, tails to move west. If in borderline county, wrap around (alternative: move towards center?)*
2. *If the district indicated by the coin (east or west) has more voters than the present district, move there.*
3. *If the district indicated by the coin has fewer likely voters, make the decision based on a probability calculation:*
4. *calculate the probability of moving as the ratio of the number of likely voters in the proposed district, to the number of voters in the current district:*
5. *Pr[move] = voters in indicated district/voters in present district*
6. *Take a random sample between 0 and 1.*
7. *If the value of the random sample is between 0 and the probability of moving, move. Otherwise, stay put.*

pacman::p\_load(wrappedtools, tidyverse, ggrepel, ggforce,   
 ggnewscale, ggtext)  
  
set.seed(1012)  
counties <- tibble(County = LETTERS[1:9],  
 Population\_defined=seq(from=10^4,   
 to= 9\*10^4,   
 by=10^4),  
 Population\_sampled = runif(n = 9,  
 min = 10^4,  
 max = 9\*10^4) |>   
 roundR(level = 2,  
 textout = F,  
 smooth = T))  
pop\_selected <- 'Population\_sampled'  
ggplot(counties,aes(County,.data[[pop\_selected]]))+  
 geom\_col()



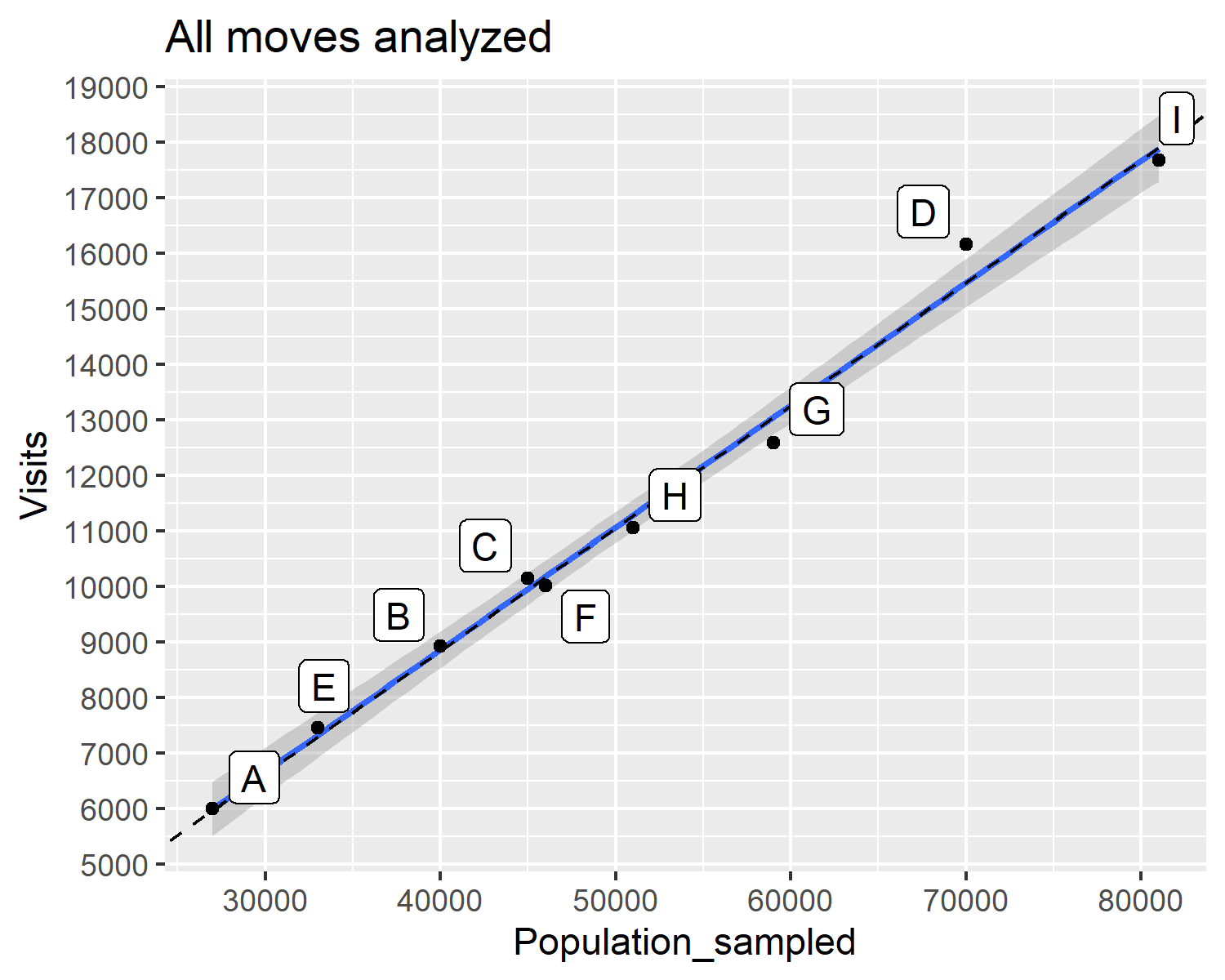
move\_selection <- function(.counties=counties,   
 current\_county,  
 which\_population=pop\_selected) {  
 coinresult <- sample(x = c(1,-1),  
 size = 1)  
 # if(current\_county==1) {  
 # coinresult <- 1  
 # }  
 # if(current\_county==nrow(.counties)) {  
 # coinresult <- -1  
 # }  
 next\_county <- current\_county+coinresult  
 if(next\_county==0) {next\_county <- nrow(.counties)}  
 if(next\_county>nrow(counties)) {next\_county <- 1}  
 population\_ratio <- .counties[[next\_county,which\_population]] /  
 .counties[[current\_county,which\_population]]  
 if(runif(n = 1,0,1)>population\_ratio){  
 next\_county <- current\_county  
 }  
 return(next\_county)  
}  
  
n\_moves <- 10^5  
n\_burnin <- 10^3  
start\_county <- 5

moves <- tibble(move=seq\_len(n\_moves),  
 position=NA\_integer\_)  
moves$position[1] <- start\_county

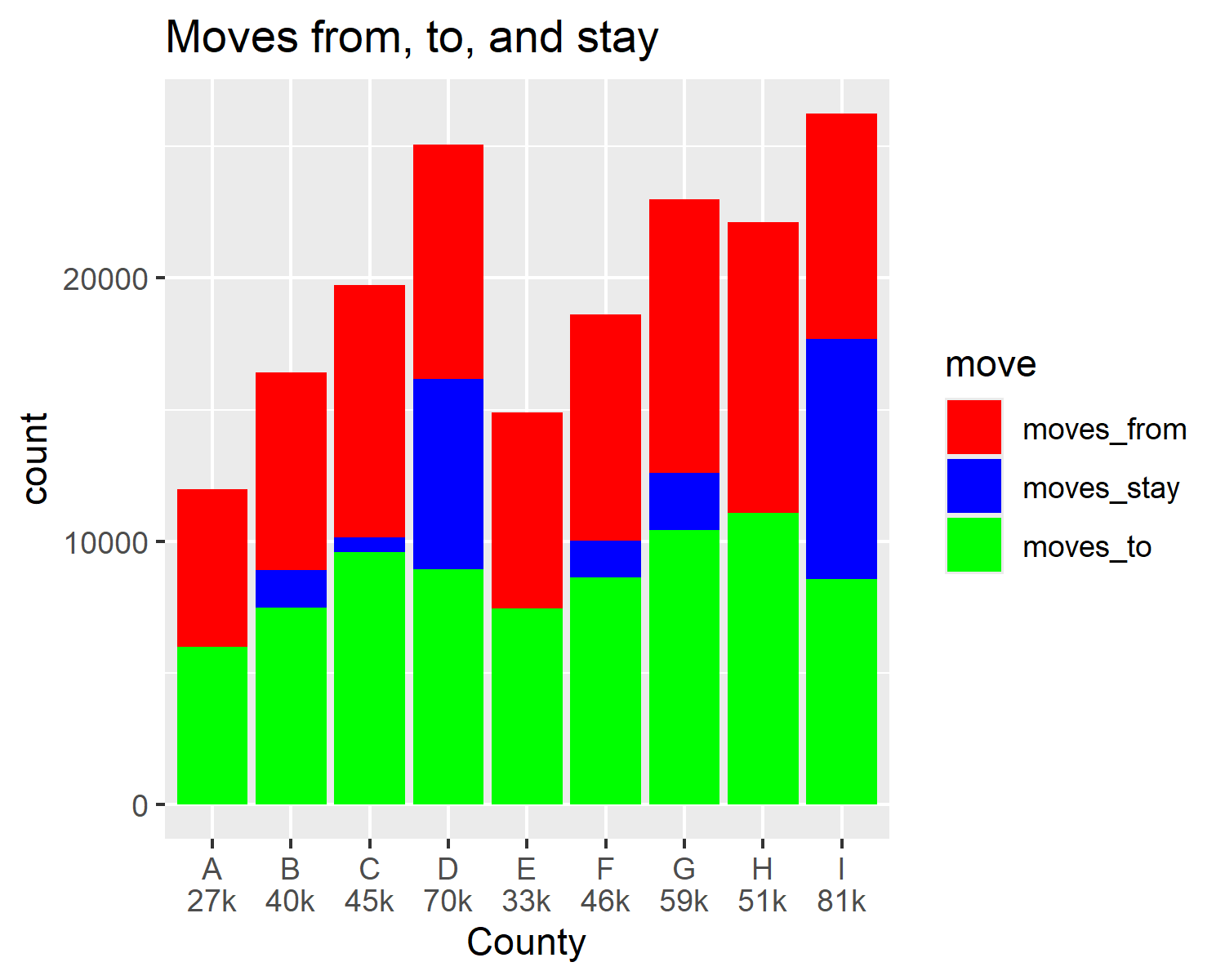
set.seed(1210)  
tictoc::tic('here we go...')  
for(step\_i in 2:n\_moves){  
 moves$position[step\_i] <-   
 move\_selection(current\_county = moves$position[step\_i-1])  
}  
tictoc::toc()

here we go...: 47.64 sec elapsed

visits <- moves |>   
 group\_by(position) |>   
 summarise(Visits=n()) |>   
 ungroup() |>   
 mutate(County = LETTERS[position]) |>   
 select(-position) |>   
 full\_join(counties)  
ggplot(visits,aes(.data[[pop\_selected]],Visits))+  
 geom\_smooth(method='lm')+  
 geom\_abline(intercept = 0,  
 slope = n\_moves/sum(counties[[pop\_selected]]),  
 linetype=2)+  
 geom\_point()+  
 geom\_label\_repel(aes(label=County),nudge\_x = 0, nudge\_y = 100)+  
 scale\_shape\_manual(values=LETTERS, guide = NULL)+  
 scale\_x\_continuous(breaks=seq(0,10^5,10^4))+  
 scale\_y\_continuous(breaks=seq(0,10^5,10^3))+  
 ggtitle('All moves analyzed')



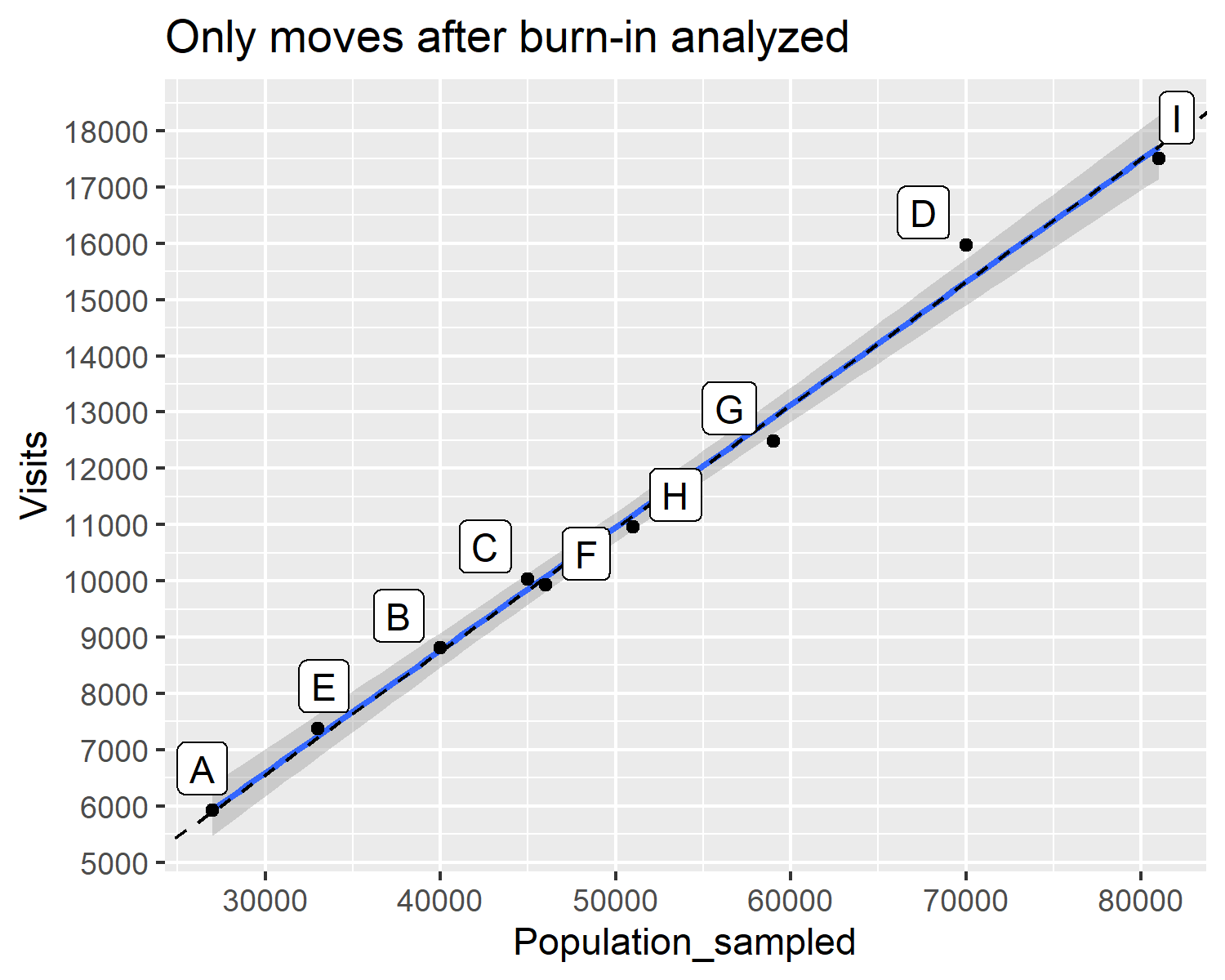
visits <- moves |>  
 filter(move>n\_burnin) |>   
 group\_by(position) |>   
 summarise(Visits=n()) |>   
 ungroup() |>   
 mutate(County = LETTERS[position]) |>   
 select(-position) |>   
 full\_join(counties)  
moves\_from\_to <- expand.grid(1:9,1:9) |>   
 as\_tibble() |>   
 rename(from=Var1,  
 to=Var2) |>   
 filter(abs(from-to)<2|abs(from-to)==8) |>   
 mutate(count=0,  
 start=LETTERS[from],  
 stop=LETTERS[to])  
for(move\_i in seq\_len(nrow(moves\_from\_to))){  
 moves\_from\_to$count[move\_i] <-   
 sum(moves$position[-nrow(moves)]==moves\_from\_to$from[move\_i] &  
 moves$position[-1]==moves\_from\_to$to[move\_i])  
}  
moves\_to <- moves\_from\_to |>   
 filter(to!=from)|> group\_by(to) |> summarize(moves\_to=sum(count)) |>   
 mutate(County=LETTERS[to])  
moves\_from <- moves\_from\_to |>   
 filter(to!=from)|> group\_by(from) |> summarize(moves\_from=sum(count)) |>   
 mutate(County=LETTERS[from])  
moves\_stay <- moves\_from\_to |>   
 filter(to==from)|> group\_by(from) |> summarize(moves\_stay=sum(count)) |>   
 mutate(County=LETTERS[from])  
moves\_from\_to\_stay <-   
 full\_join(moves\_from,moves\_to) |>   
 full\_join(moves\_stay) |>   
 full\_join(counties |> select(-Population\_defined)) |>   
 select(-from,-to) |>   
 pivot\_longer(cols=c(moves\_from,moves\_to,moves\_stay),  
 names\_to='move',  
 values\_to='count') |>   
 mutate(County=paste0(County,"\n",  
 round(Population\_sampled/1000),  
 "k"  
 ))   
  
ggplot(moves\_from\_to\_stay,aes(County,count, fill=move))+  
 geom\_col()+  
 scale\_fill\_manual(values=c('red','blue','green'))+  
 ggtitle('Moves from, to, and stay')



cat("<br>\n\n")

<br>

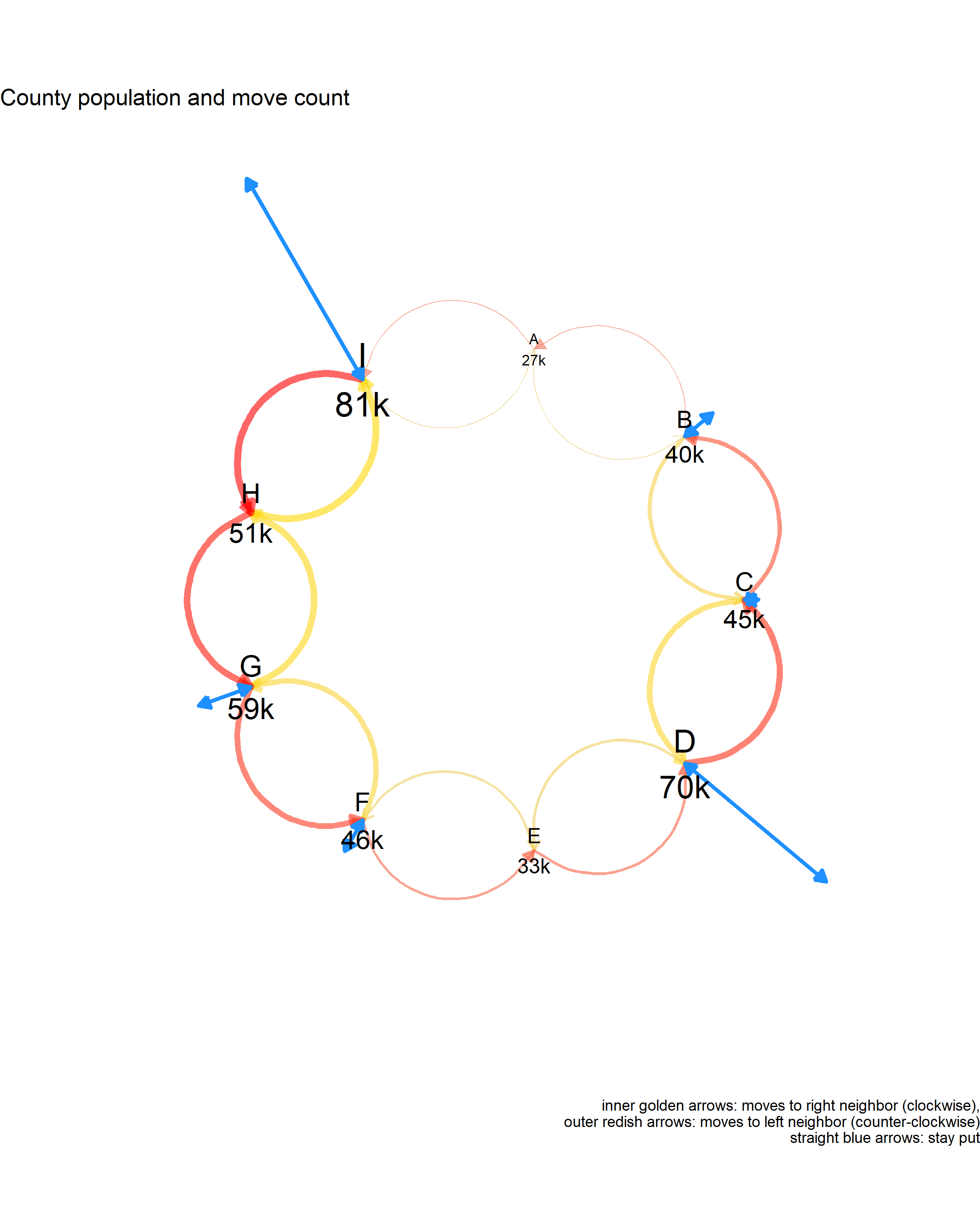
ggplot(visits,aes(.data[[pop\_selected]],Visits))+  
 geom\_smooth(method='lm')+  
 geom\_abline(intercept = 0,  
 slope = (n\_moves-n\_burnin)/sum(counties[[pop\_selected]]),  
 linetype=2)+  
 geom\_point()+  
 geom\_label\_repel(aes(label=County),nudge\_x = 0, nudge\_y = 100)+  
 scale\_shape\_manual(values=LETTERS, guide = NULL)+  
 scale\_x\_continuous(breaks=seq(0,10^5,10^4))+  
 scale\_y\_continuous(breaks=seq(0,10^5,10^3))+  
 ggtitle('Only moves after burn-in analyzed')



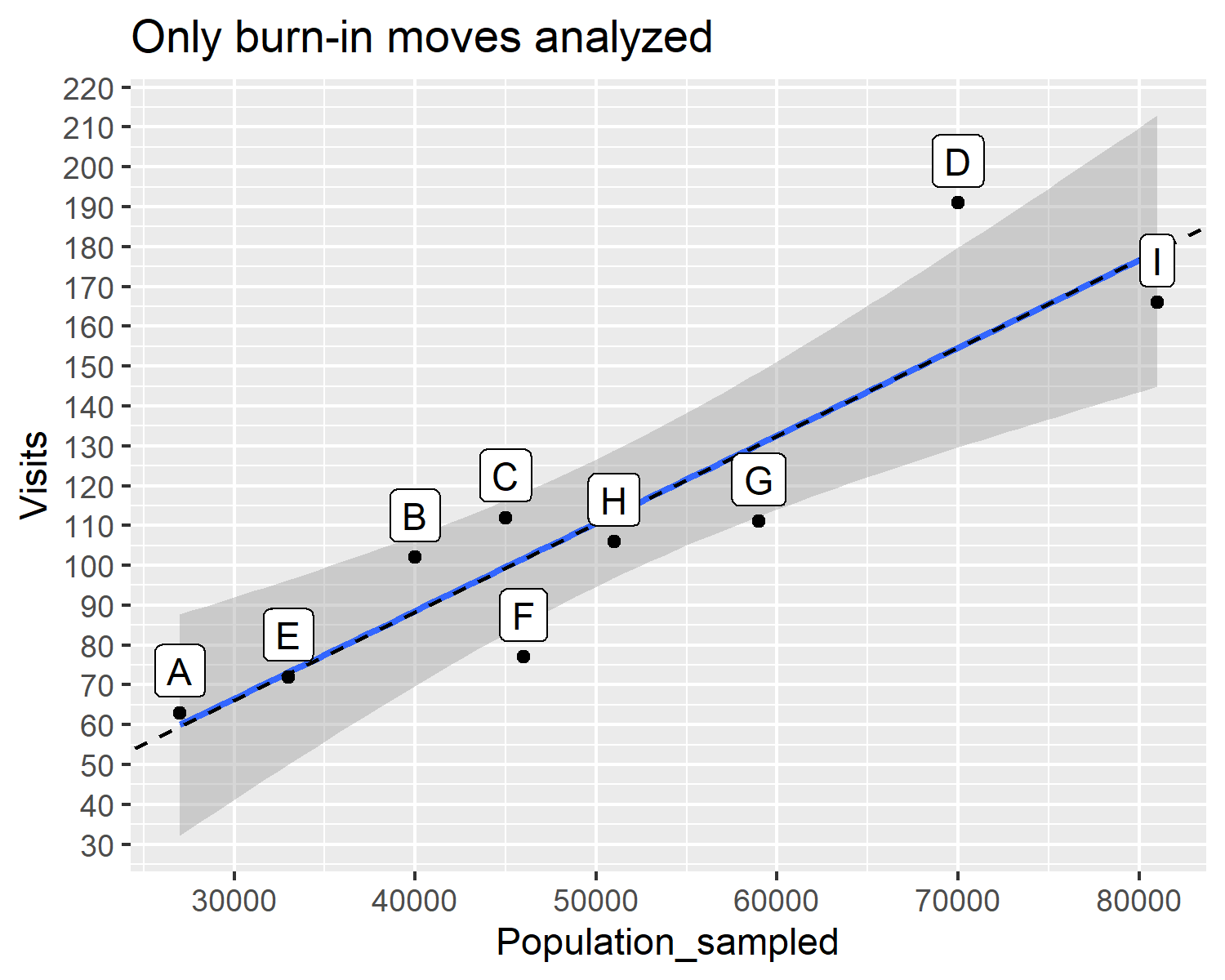
cat("<br>\n\n")

<br>

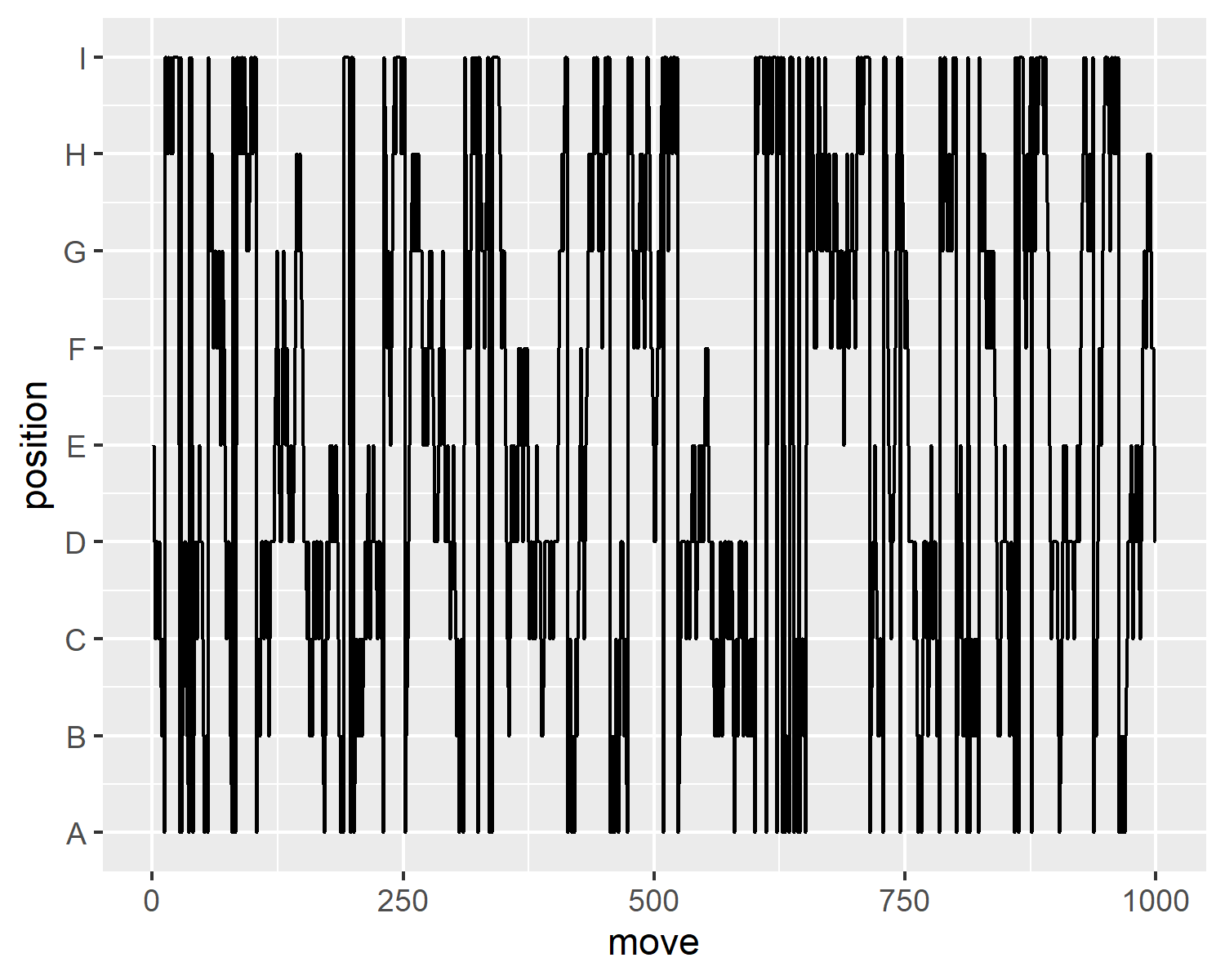
# circle for county plot  
# Define circle aesthetics  
theta <- seq(0, 2\*pi, length.out = 10)[c(4:9,1:3)] |>   
 rev()  
# Create sequence for angles (0 to 2\*pi) with 5 equally spaced points  
radius <- 1 # Set radius of the circle  
  
# Create data frame with circle coordinates and labels  
circle\_data <- tibble(  
 angle=theta,  
 x = radius \* cos(theta),  
 y = radius \* sin(theta),  
 label =LETTERS[1:9] # Assign letters A to E as labels  
) |>   
 full\_join(visits, by=c('label'='County')) |>   
 mutate(plotlabel=paste0(label,"\n",  
 round(Population\_sampled/1000),  
 "k"  
 ))   
arrow\_data <-   
 moves\_from\_to |>   
 full\_join(circle\_data |>   
 select(x:label, angle), by=c('start'='label')) |>   
 rename(from\_x="x",from\_y="y") |>   
 full\_join(circle\_data |>   
 select(x:label), by=c('stop'='label')) |>   
 rename(to\_x="x",to\_y="y") |>   
 mutate(x\_end=from\_x+count\*10/n\_moves\*cos(angle),  
 y\_end=from\_y+count\*10/n\_moves\*sin(angle),  
 count=case\_when((from<to & !(from==1 & to==9)) | (from==9 & to==1)~-count,  
 .default=count))   
  
   
# Createggplot with circle and labels  
# ggplot(circle\_data, aes(x = x, y = y)) +  
# # geom\_point(aes(size = Visits), shape=1) + # Increase point size for better visibility  
# geom\_circle(aes(r=1,x0=0,y0=0),  
# color="darkorange2")+  
# geom\_curve(data=arrow\_data |>   
# filter(from<to),  
# aes(x=from\_x,y=from\_y,  
# xend=to\_x,yend=to\_y,  
# linewidth=count),  
# arrow=arrow(length=unit(0.1,"inches")),  
# curvature=-1.0,  
# color="darkolivegreen")+  
# geom\_curve(data=arrow\_data |>   
# filter(from>to),  
# aes(x=from\_x,y=from\_y,  
# xend=to\_x,yend=to\_y,  
# linewidth=count),  
# arrow=arrow(length=unit(0.1,"inches")),  
# angle=90,  
# curvature=-.75,  
# color="dodgerblue")+  
# geom\_text(aes(label = plotlabel, size=.data[[pop\_selected]]),  
# hjust = 0.5, vjust = 0.5) + # Adjust text position slightly  
# scale\_size\_continuous(range=c(3,7)) + # Set size of labels  
# scale\_linewidth\_continuous(range=c(.5,3)) + # Set size of labels  
# coord\_fixed(xlim = c(-radius - radius/5, radius + radius/5), ylim = c(-radius - radius/5, radius + radius/5)) + # Set axis limits slightly bigger than radius  
# labs(title = "County population and move count", x = "", y = "",  
# caption = "inner arrows: moves to left neighbor,\nouter arrows: moves to right neighbor") + # Add title and remove axis labels  
# guides(size="none", linewidth="none")+  
# theme\_void() # Remove background gridlines  
# cat("<br>\n\n")  
  
# Create ggplot with circle and labels  
ggplot(circle\_data, aes(x = x, y = y)) +  
 # geom\_point(aes(size = Visits), shape=1) + # Increase point size for better visibility  
 # geom\_circle(aes(r=1,x0=0,y0=0),  
 # color="darkorange2")+  
 geom\_curve(data=arrow\_data |>   
 filter((from<to & !(to==9 & from==1)) | (to==1 & from==9)),  
 aes(x=from\_x,y=from\_y,  
 xend=to\_x,yend=to\_y,  
 color=count, linewidth=abs(count)),  
 arrow=arrow(length=unit(0.1,"inches")),  
 curvature=.75,  
 alpha=.6)+#, linewidth=1.5,)+  
 scale\_colour\_gradient2(low="gold", mid="grey",high="red")+  
 # scale\_color\_gradient("move count", low = "gold",high = "gold4") +  
 geom\_curve(data=arrow\_data |>   
 filter(from>to &!(from==9 & to==1) | (to==9 & from==1)),  
 aes(x=from\_x,y=from\_y,  
 xend=to\_x,yend=to\_y,  
 color=count, linewidth=abs(count)),  
 arrow=arrow(length=unit(0.1,"inches"),  
 type="closed"),  
 # angle=90,#linewidth=1.5,  
 curvature=.75,  
 alpha=.6)+  
 geom\_segment(data=arrow\_data |>   
 filter(from==to,count>0),  
 aes(x=from\_x,y=from\_y,  
 xend=x\_end,  
 yend=y\_end),  
 arrow=arrow(length=unit(0.1,"inches"),  
 ends = "both",  
 type="closed"),  
 color='dodgerblue', linewidth=1.2)+  
 geom\_text(aes(label = plotlabel, size=.data[[pop\_selected]]),  
 hjust = 0.5, vjust = 0.5) + # Adjust text position slightly  
 scale\_size\_continuous(range=c(3,7)) + # Set size of labels  
 scale\_linewidth\_continuous(range = c(.25,2))+# Set size of labels  
 coord\_fixed(xlim = c(-radius \* 1.75, radius \* 1.75),  
 ylim = c(-radius \* 1.75, radius \* 1.75)) + # Set axis limits slightly bigger than radius  
 labs(title = "County population and move count", x = "", y = "",  
 caption = "inner golden arrows: moves to right neighbor (clockwise),\nouter redish arrows: moves to left neighbor (counter-clockwise)\nstraight blue arrows: stay put") + # Add title and remove axis labels  
 guides(size="none", linewidth="none", color="none")+  
 theme\_void() # Remove background gridlines



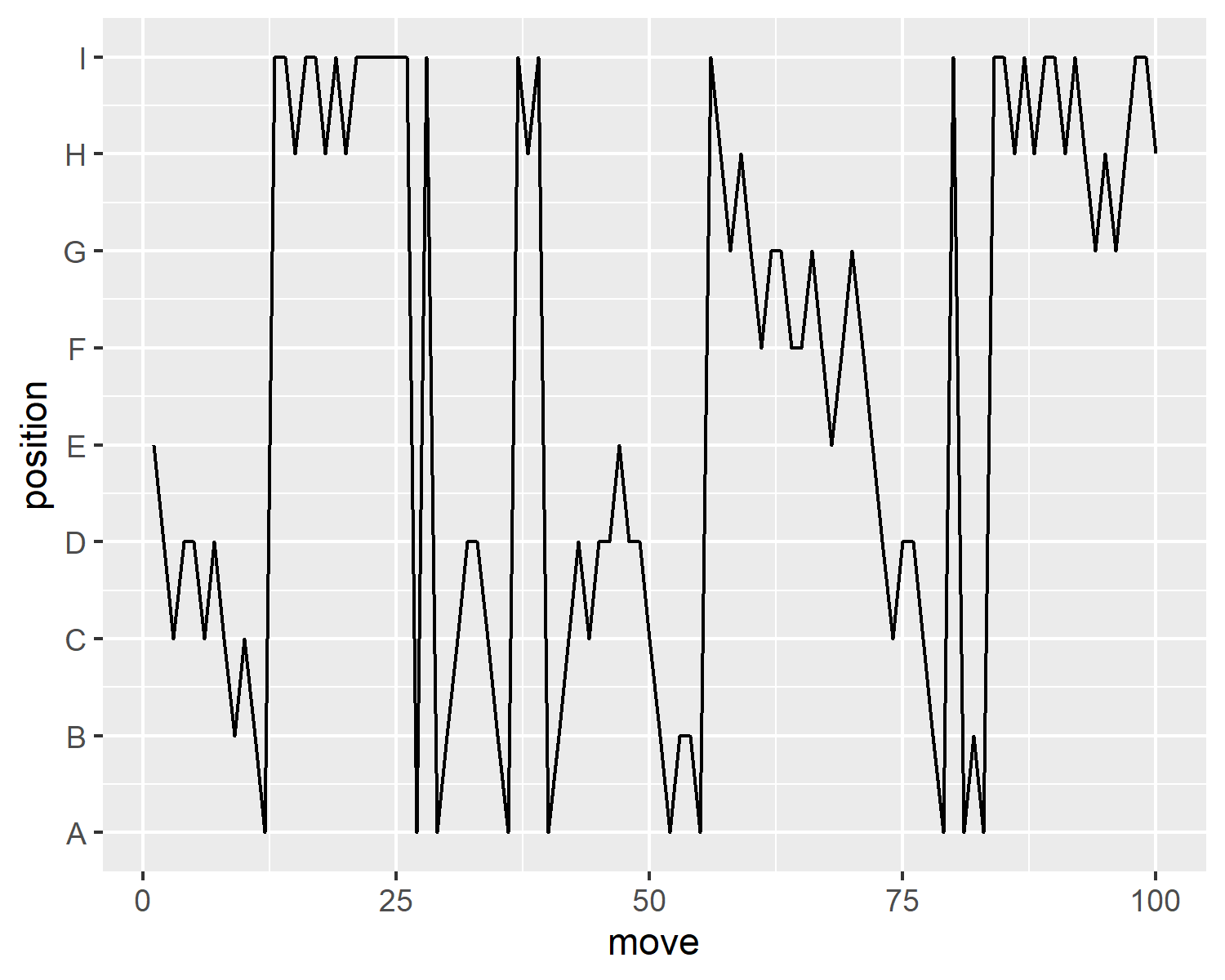
visits <- moves |>  
 filter(move<=n\_burnin) |>   
 group\_by(position) |>   
 summarise(Visits=n()) |>   
 ungroup() |>   
 mutate(County = LETTERS[position]) |>   
 select(-position) |>   
 full\_join(counties)  
ggplot(visits,aes(.data[[pop\_selected]],Visits))+  
 geom\_smooth(method='lm')+  
 geom\_abline(intercept = 0,  
 slope = n\_burnin/sum(counties[[pop\_selected]]),  
 linetype=2)+  
 geom\_point()+  
 geom\_label\_repel(aes(label=County),nudge\_x = 0, nudge\_y = 10)+  
 scale\_shape\_manual(values=LETTERS, guide = NULL)+  
 scale\_x\_continuous(breaks=seq(0,10^5,10^4))+  
 scale\_y\_continuous(breaks=seq(0,10^5,10^1))+  
 ggtitle('Only burn-in moves analyzed')



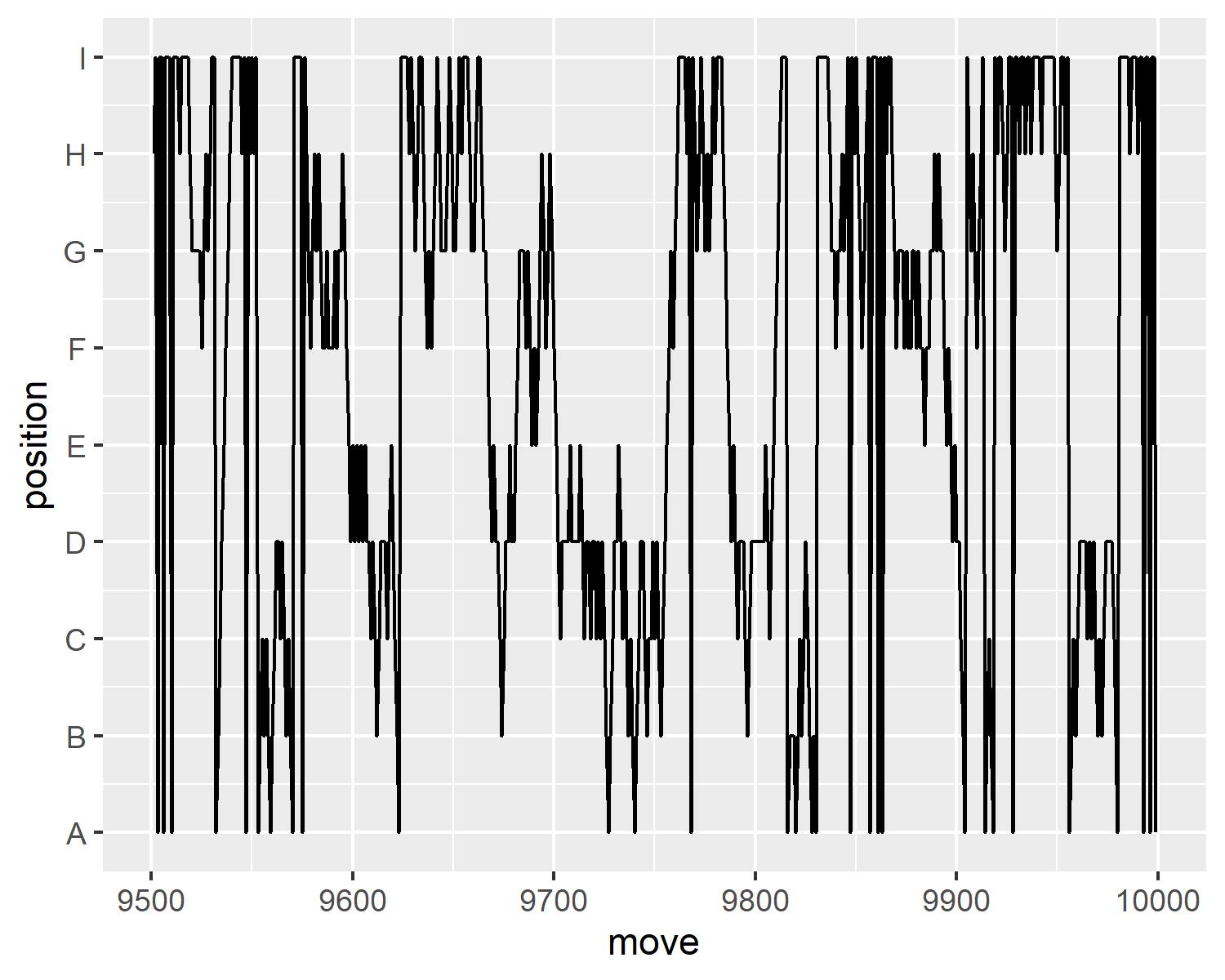
moves |>   
 filter(move<=n\_burnin) |>  
 ggplot(aes(move,position))+  
 # geom\_point()+  
 geom\_line()+  
 scale\_y\_continuous(breaks=1:9,  
 labels = LETTERS[1:9])



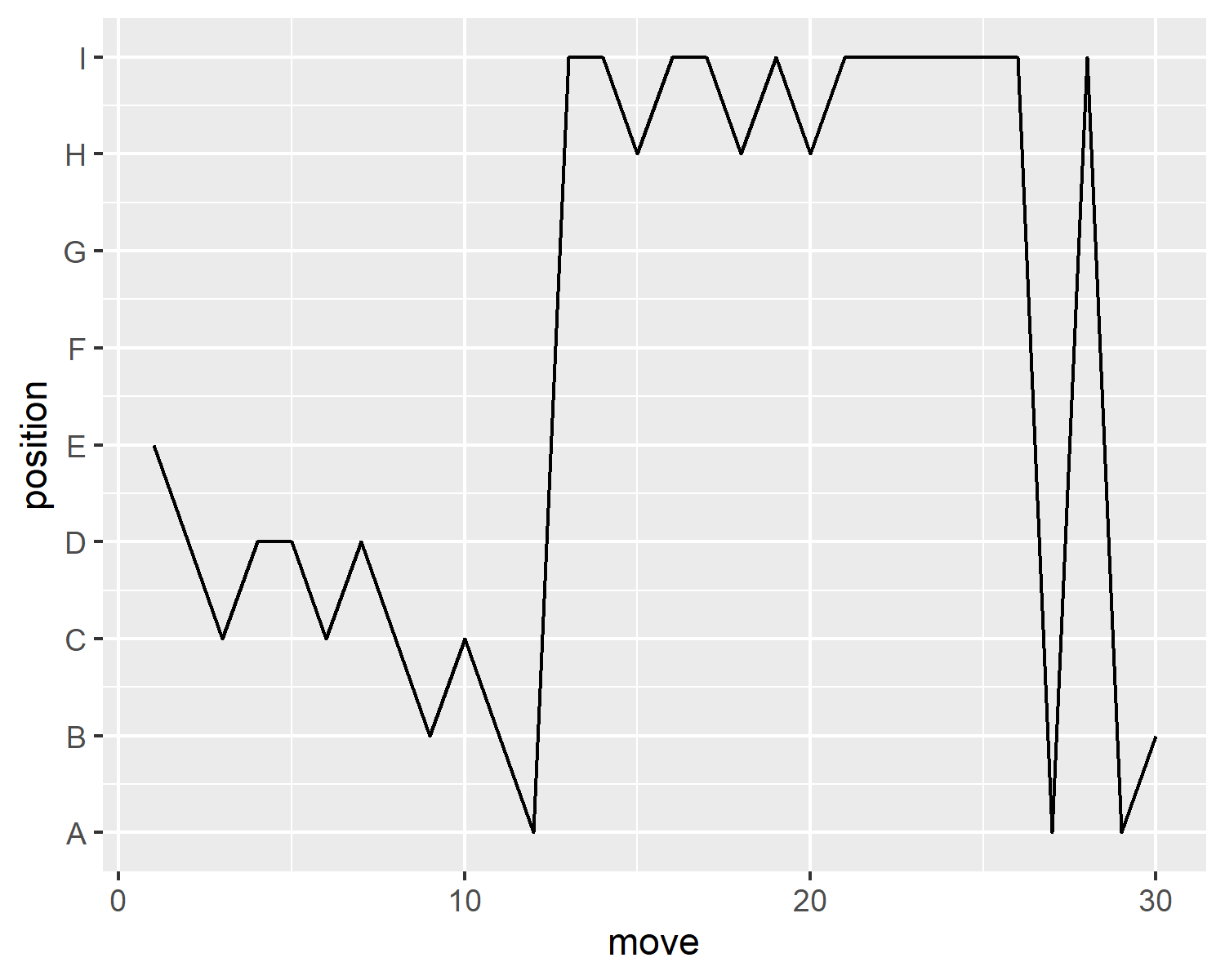
moves |>   
 filter(move<=100) |>  
 ggplot(aes(move,position))+  
 # geom\_point()+  
 geom\_line()+  
 scale\_y\_continuous(breaks=1:9,  
 labels = LETTERS[1:9])



moves |>   
 filter(move>9500, move<10000) |>  
 ggplot(aes(move,position))+  
 # geom\_point()+  
 geom\_line()+  
 scale\_y\_continuous(breaks=1:9,  
 labels = LETTERS[1:9])



moves |>   
 # filter(move<=n\_burnin) |>  
 ggplot(aes(move,position))+  
 # geom\_point()+  
 geom\_line()+  
 scale\_x\_continuous(limits = c(1,30))+  
 scale\_y\_continuous(breaks=1:9,  
 labels = LETTERS[1:9])



moves |>   
 # filter(move<=n\_burnin) |>  
 ggplot(aes(move,position))+  
 # geom\_point()+  
 geom\_line()+  
 # scale\_x\_continuous(limits = c(1,30))+  
 scale\_y\_continuous(breaks=1:9,  
 labels = LETTERS[1:9])+  
 coord\_cartesian(xlim=c(1,30))

