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)  
  
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 = 3,  
 textout = F,  
 smooth = T))  
pop\_selected <- 'Population\_sampled'  
ggplot(counties,aes(County,.data[[pop\_selected]]))+  
 geom\_col()

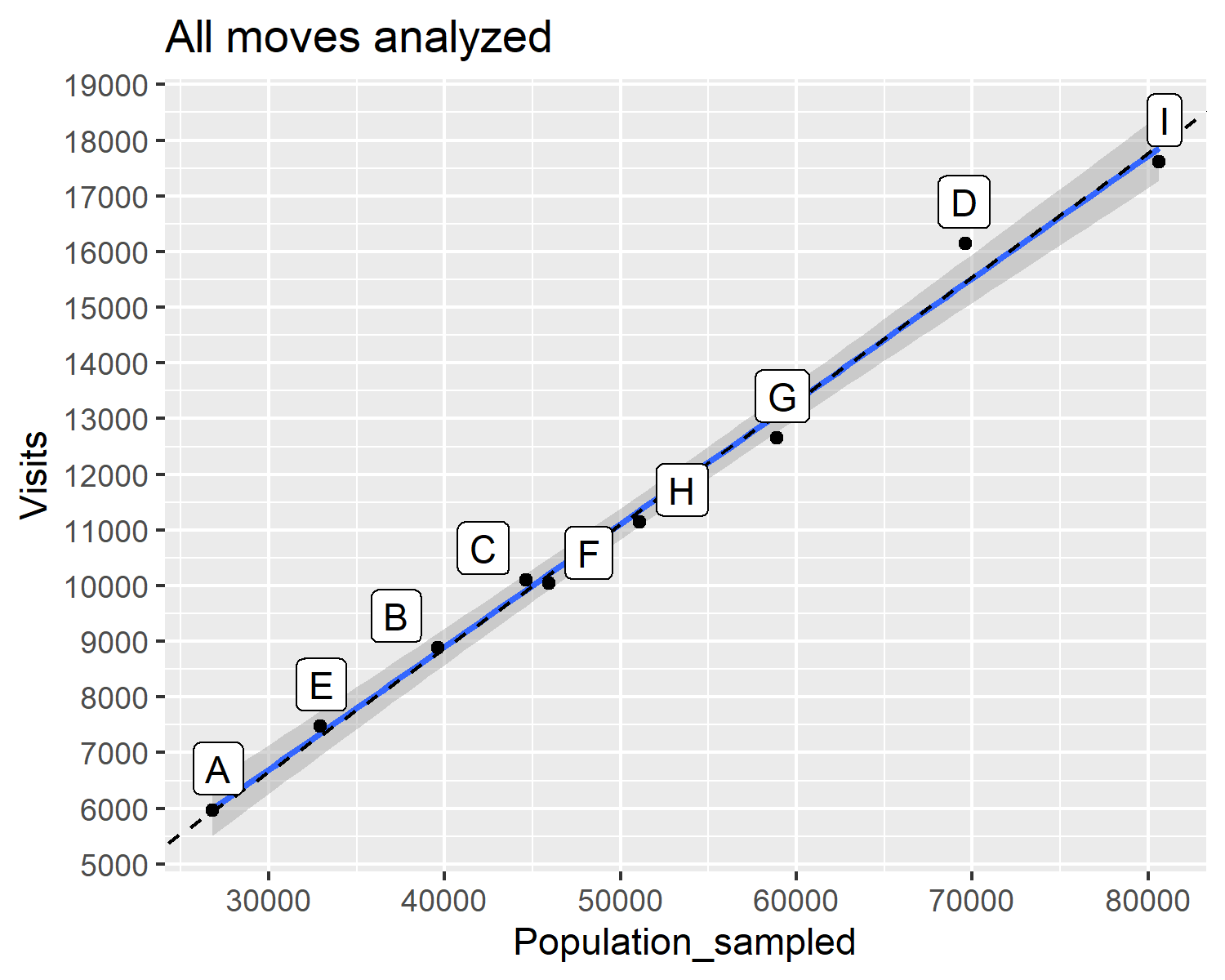


move\_selection <- function(.counties=counties,   
 current\_county,  
 which\_population=pop\_selected) {  
 coinresult <- sample(x = c(1,-1),  
 size = 1)  
 # if(current\_county==1) {  
 # coinresult <- 1  
 # }  
 # if(current\_county==nrow(.counties)) {  
 # coinresult <- -1  
 # }  
 next\_county <- 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

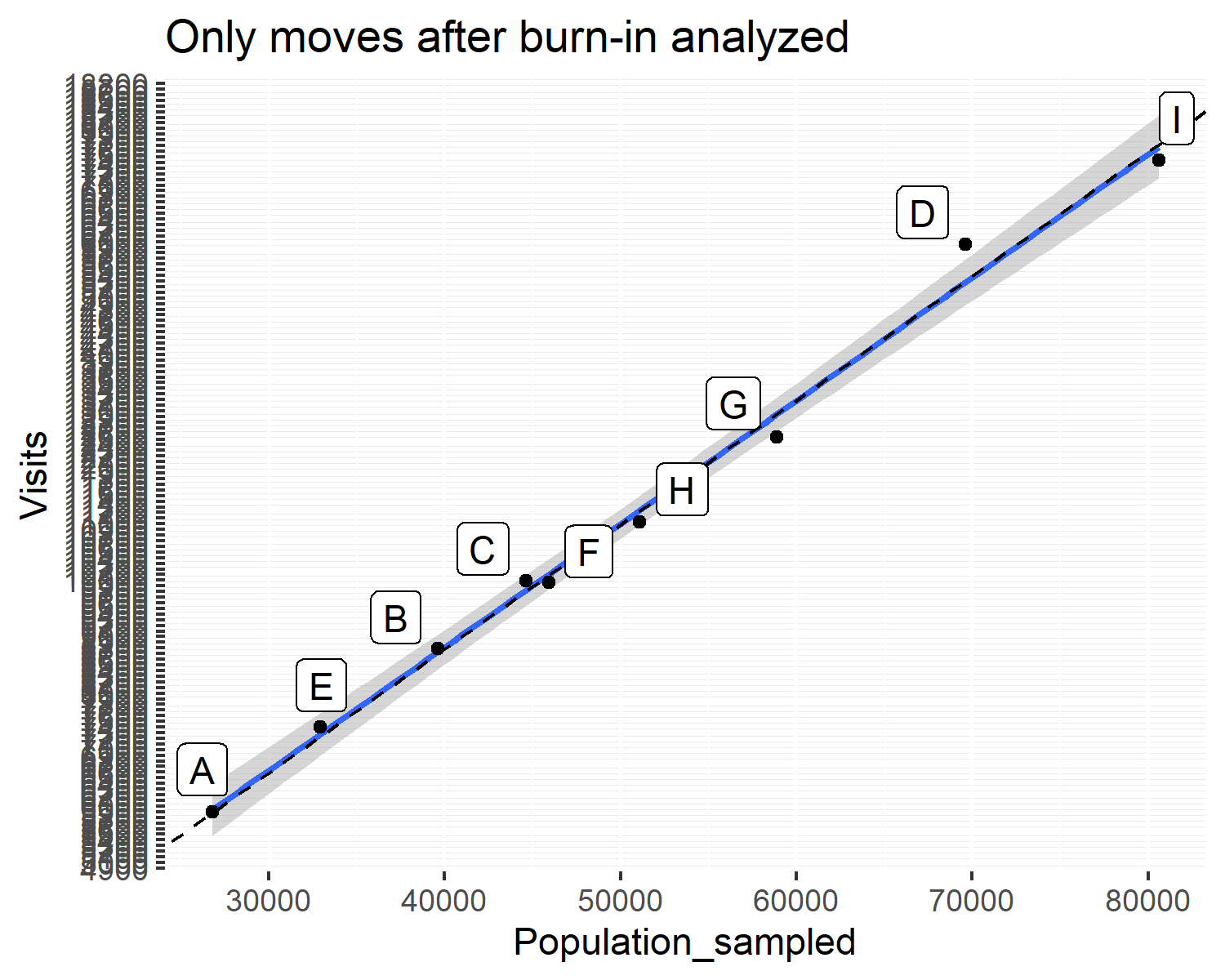
moves <- tibble(move=seq\_len(n\_moves),  
 position=NA\_integer\_)  
moves$position[1] <- start\_county

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

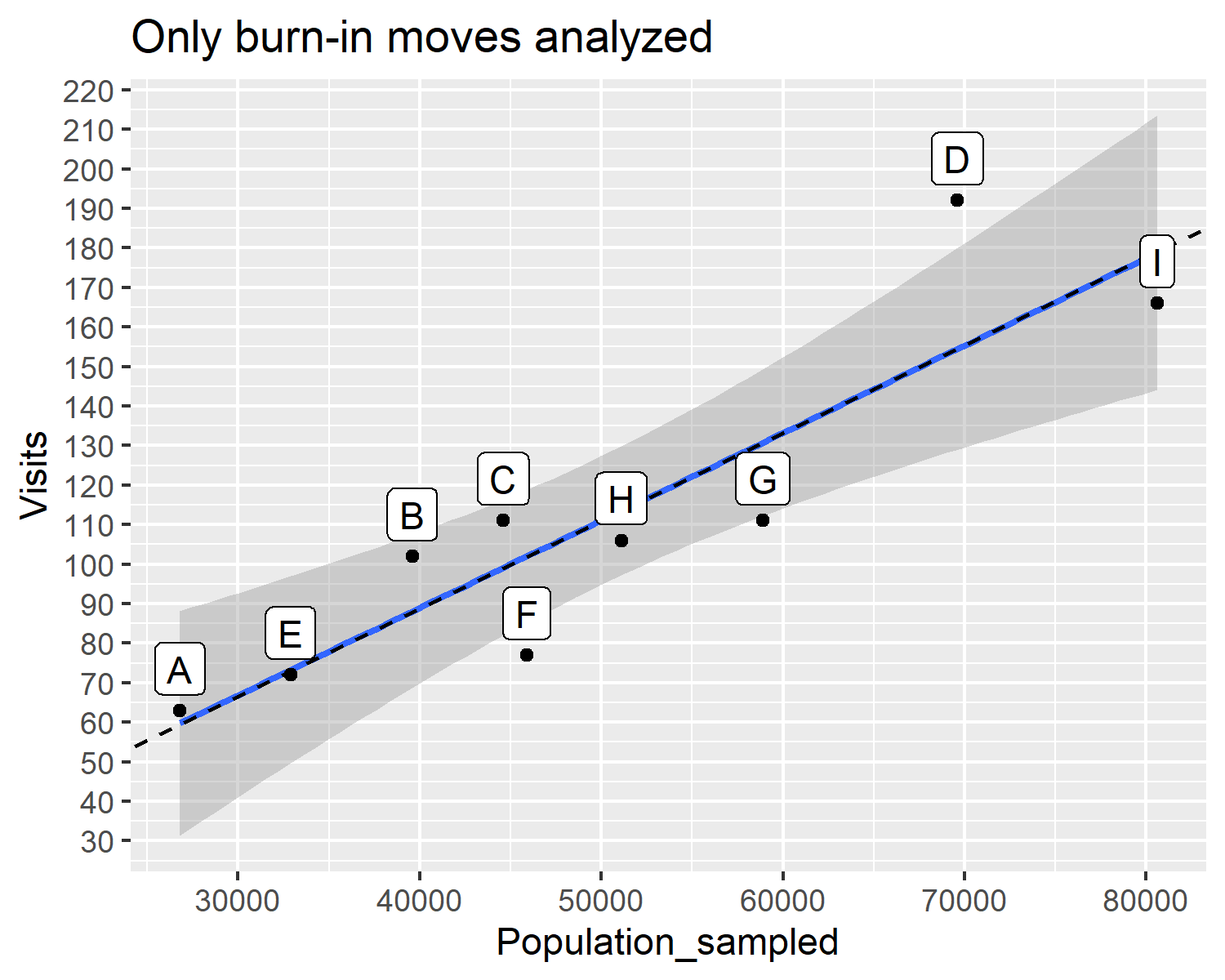
visits <- moves |>   
 group\_by(position) |>   
 summarise(Visits=n()) |>   
 ungroup() |>   
 mutate(County = LETTERS[position]) |>   
 select(-position) |>   
 full\_join(counties)  
ggplot(visits,aes(.data[[pop\_selected]],Visits))+  
 geom\_smooth(method='lm')+  
 geom\_abline(intercept = 0,  
 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')



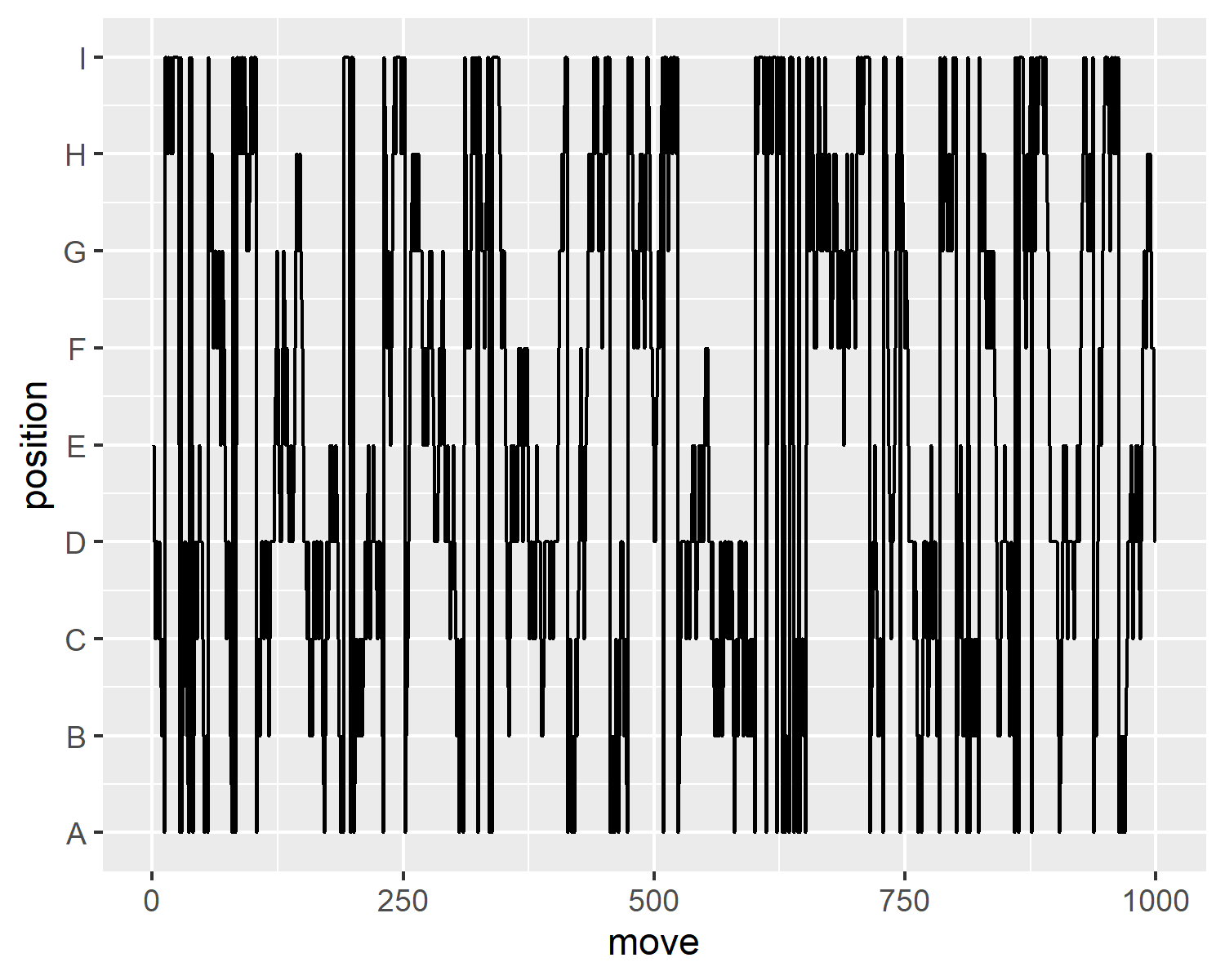
visits <- moves |>  
 filter(move>n\_burnin) |>   
 group\_by(position) |>   
 summarise(Visits=n()) |>   
 ungroup() |>   
 mutate(County = LETTERS[position]) |>   
 select(-position) |>   
 full\_join(counties)  
ggplot(visits,aes(.data[[pop\_selected]],Visits))+  
 geom\_smooth(method='lm')+  
 geom\_abline(intercept = 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(0,10^5,10^2))+  
 ggtitle('Only moves after burn-in analyzed')



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])

