

Estudos eleitorais com R

Willber Nascimento

Doutorando DCP/UFPE

nascimentowillber@gmail.com
(<mailto:nascimentowillber@gmail.com>)

- Eleições limpas, livres e competitivas são o cerne do que entendemos por modelo democrático representativo.
- No Brasil celebramos a festa da democracia a cada dois anos:



- **Eu celebro a festa dos dados abertos!**
- Dados atualizados de dois em dois anos:



- Textões para Facebook e Twitter
- Artigos Qualis A



- Os estudos eleitorais possuem uma quantidade enorme de

conceitos / medidas

- Escolha alguns indicadores importantes
 - Número competidores
 - Flutuação do apoio
 - Distorções eleitorais
 - Presença partidária
- **Mas como podemos usar esses dados de forma precisa, comparativa e replicável para propósitos variados?**
 - Use R!
 - electionsBR
 - dplyr
 - ggplot

Número de candidatos (Fragmentação)

Carregando os dados

In [2]:

```
base_uf_partido <- readRDS('../base_uf_partido.Rda')  
head(base_uf_partido[,c(1,2,5,6,7)])
```

ANO_ELEICAO	SIGLA_UF	SIGLA_PARTIDO	TOTAL_V
1998	AC	PP	23923
1998	AC	PDT	7969
1998	AC	PT	39090
1998	AC	MDB	26417
1998	AC	PSTU	17
1998	AC	PR	349

In [3]:

```
# Descrevendo o N de candidatos
```

```
base_uf_partido %>%  
  filter(DESCRICA0_CARGO=='DEPUTADO FEDERAL') %>%  
  group_by(ANO_ELEICAO, SIGLA_UF) %>% summarise(n = n()) %>% ungro  
up() %>% group_by(ANO_ELEICAO) %>%  
  summarise(Média = mean(n), `Desvio Padrão` = sd(n))
```

ANO_ELEICAO	Média	Desvio Padrão
1998	18.92593	4.874785
2002	23.03704	3.578585
2006	23.81481	3.051318
2010	22.55556	2.485857
2014	28.74074	2.595745
2018	32.59259	1.670082

Número efetivo de partidos

- Desenvolvido por Laakso e Taagepera (1979) para mensurar de forma mais precisa a forma dos sistemas partidários.
 - Quantos partidos compõe o sistema:
 - Partidos competitivos
 - Medida numérica
- Como calcular?

$$NEP = \frac{1}{\sum_{i=1}^n p_i^2}$$

Onde:

- p_i^2 é a proporção de votos ao quadrado do partido i em um determinado distrito.

- O que ele mede?
 - O número hipotético de partidos de tamanhos iguais.

Fracionalização partidária

- Desenvolvido por Rae (1967).
 - O que ele mede?
 - A fracionalização do apoio eleitoral entre os partidos. Intuitivamente: a probabilidade de que dois eleitores tenham votado em partidos diferentes.
- Como medir?

$$FRAC = 1 - \sum_{i=1}^n p_i^2$$

Onde:

- p_i^2 é a proporção de votos ao quadrado do partido i em um determinado distrito.

Aplicando NEP para Pernambuco

In [4]:

```
pe_2018 <-  
  base_uf_partido[base_uf_partido$ANO_ELEICAO==2018 &  
    base_uf_partido$DESCRICAO_CARGO=='DEPUTADO FEDERA  
L' &  
    base_uf_partido$SIGLA_UF == 'PE' &  
    base_uf_partido$PROP_VOTOS > 0,]  
  
head(pe_2018)
```

ANO_ELEICAO	SIGLA_UF	DESCRICAO_CARGO	NUM
2018	PE	DEPUTADO FEDERAL	10
2018	PE	DEPUTADO FEDERAL	11
2018	PE	DEPUTADO FEDERAL	12
2018	PE	DEPUTADO FEDERAL	13
2018	PE	DEPUTADO FEDERAL	14
2018	PE	DEPUTADO FEDERAL	15

In [5]:

```
# A fórmula  
  
1/sum((pe_2018$PROP_VOTOS^2))
```

12.04818131198

Pernambuco: Governador e várias eleições

In [6]:

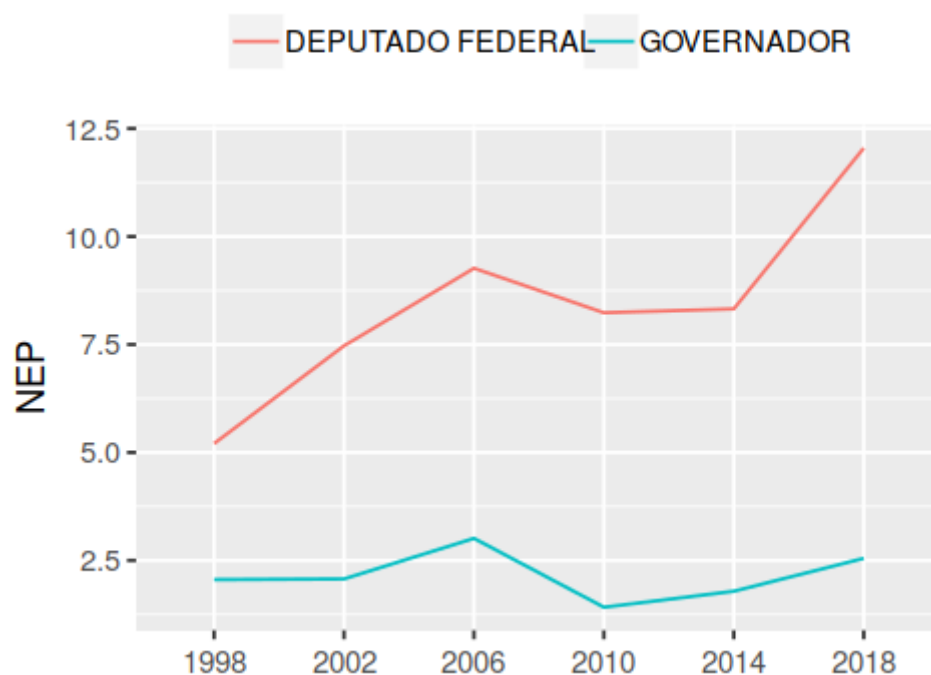
```
base_uf_partido %>% group_by(ANO_ELEICAO, DESCRICAO_CARGO) %>%  
  filter(SIGLA_UF=='PE' & PROP_VOTOS>0) %>%  
  summarise(  
    NEP = 1/sum((PROP_VOTOS^2))  
  )
```

ANO_ELEICAO	DESCRICAO_CARGO	NEP
1998	DEPUTADO FEDERAL	5.205707
1998	GOVERNADOR	2.054779
2002	DEPUTADO FEDERAL	7.469686
2002	GOVERNADOR	2.070725
2006	DEPUTADO FEDERAL	9.265330
2006	GOVERNADOR	3.010343
2010	DEPUTADO FEDERAL	8.235153
2010	GOVERNADOR	1.415497
2014	DEPUTADO FEDERAL	8.324500
2014	GOVERNADOR	1.785610
2018	DEPUTADO FEDERAL	12.048181
2018	GOVERNADOR	2.548076

In [7]:

```
options(repr.plot.width=4, repr.plot.height=3)

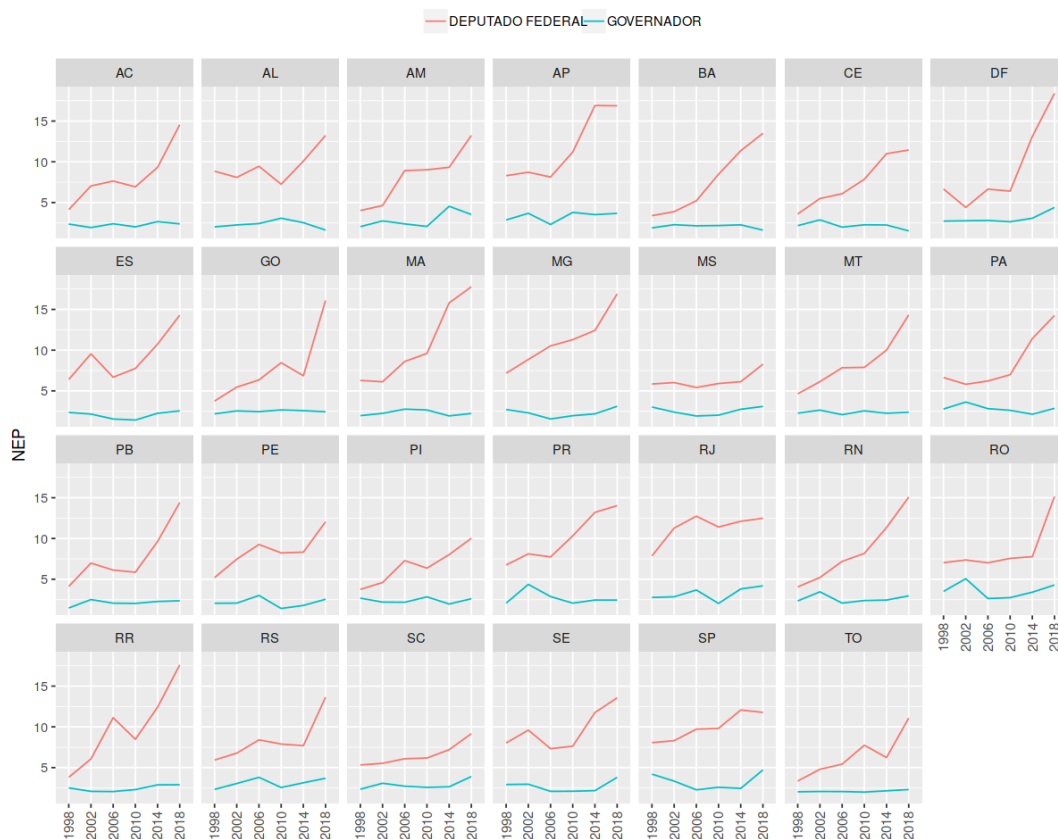
base_uf_partido %>% group_by(ANO_ELEICA0, DESCRICA0_CARGO) %>%
  filter(SIGLA_UF=='PE' & PROP_VOTOS>0) %>%
  summarise(
    NEP = 1/sum((PROP_VOTOS^2))
  ) %>% ggplot(., aes(factor(ANO_ELEICA0), NEP, group=DESCRICA0_CARGO, fill=DESCRICA0_CARGO, colour=DESCRICA0_CARGO))+
    geom_line()+
    theme(axis.text.x=element_text(size=9, vjust=0.5),
          axis.title.x=element_blank(),
          legend.position = 'top',
          legend.title = element_blank())
```



Brasil: NEP 1998-2018

In [8]:

```
options(repr.plot.width=10, repr.plot.height=8)
base_uf_partido %>% group_by(ANO_ELEICAO, SIGLA_UF, DESCRICAO_CARGO)
%>%
  filter(PROP_VOTOS>0) %>%
  summarise(
    NEP = 1/sum((PROP_VOTOS^2))
  ) %>% ggplot(., aes(factor(ANO_ELEICAO), NEP, group=DESCRICAO_CARGO, fill=DESCRICAO_CARGO, colour=DESCRICAO_CARGO))+
    geom_line()+facet_wrap(~SIGLA_UF, ncol = 7)+
    theme(axis.text.x=element_text(size=9, angle=90, vjust=0.5),
          axis.title.x=element_blank(),
          legend.position = 'top',
          legend.title = element_blank())
```



Flutuação Eleitoral

Volatilidade Eleitoral

- Criado por Pedersen (1979) para medir a flutuação do apoio eleitoral dos partidos nos sistemas partidários europeus.
- Mainwaring e Torcal (2003) usam para medir institucionalização do sistema partidário.
- Como medir?

$$VT = \frac{1}{2} \sum_{i=1}^n |p_{i,t+1} - p_{i,t}|$$

Esse é um simples indicador onde:

- $p_{i,t}$ é a proporção de votos do partido i em um determinado distrito no tempo t .

Pernambuco: calculando a volatilidade

In [9]:

```
pe_2018 <-  
  base_uf_partido[base_uf_partido$ANO_ELEICAO>=2014 &  
    base_uf_partido$DESCRICAO_CARGO=='DEPUTADO FEDERA  
L' &  
    base_uf_partido$SIGLA_UF == 'PE' &  
    base_uf_partido$PROP_VOTOS > 0,]  
  
pe_2018 %>%  
  group_by(SIGLA_UF) %>%  
  dcast(., SIGLA_PARTIDO ~ ANO_ELEICAO) %>%  
  mutate(`2014` = replace_na(`2014`, 0), `2018` = replace_na(`2018`,  
0)) %>%  
  mutate(diferenca = abs(`2018` - `2014`)) %>%  
  summarise(volatilidade = (1/2*sum(diferenca)*100))
```

Using PROP_VOTOS as value column: use value.var to override.

volatilidade
34,79131

Brasil: volatilidade (1998-2018)

In [10]:

```
volatilidade <- base_uf_partido %>%
  filter(DESCRICAO_CARGO=='DEPUTADO FEDERAL') %>%
  dcast(., SIGLA_UF + SIGLA_PARTIDO ~ ANO_ELEICAO) %>%
  mutate(`1998` = replace_na(`1998`, 0),
         `2002` = replace_na(`2002`, 0),
         `2006` = replace_na(`2006`, 0),
         `2010` = replace_na(`2010`, 0),
         `2014` = replace_na(`2014`, 0),
         `2018` = replace_na(`2018`, 0)) %>%
  mutate(diferenca02 = abs(`2002` - `1998`),
         diferenca06 = abs(`2006` - `2002`),
         diferenca10 = abs(`2010` - `2006`),
         diferenca14 = abs(`2014` - `2010`),
         diferenca18 = abs(`2018` - `2014`)) %>%
  group_by(SIGLA_UF) %>%
  summarise(volatilidade02 = (1/2*sum(diferenca02)*100),
            volatilidade06 = (1/2*sum(diferenca06)*100),
            volatilidade10 = (1/2*sum(diferenca10)*100),
            volatilidade14 = (1/2*sum(diferenca14)*100),
            volatilidade18 = (1/2*sum(diferenca18)*100))

head(volatilidade)
```

Using PROP_VOTOS as value column: use value.var to override.

SIGLA_UF	volatilidade02	volatilidade06	volatilidade10
AC	51,98854	30,45730	35,98899
AL	40,13130	36,71720	41,03958
AM	33,09306	62,34962	19,91028
AP	34,74423	38,73298	31,90727
BA	12,63459	15,25283	27,65537
CE	21,31650	33,57534	24,85227

In [11]:

```
volatilidade <- volatilidade %>% melt()
volatilidade$variable <- recode(volatilidade$variable,
                                'volatilidade02' = '2002',
                                'volatilidade06' = '2006',
                                'volatilidade10' = '2010',
                                'volatilidade14' = '2014',
                                'volatilidade18' = '2018')
```

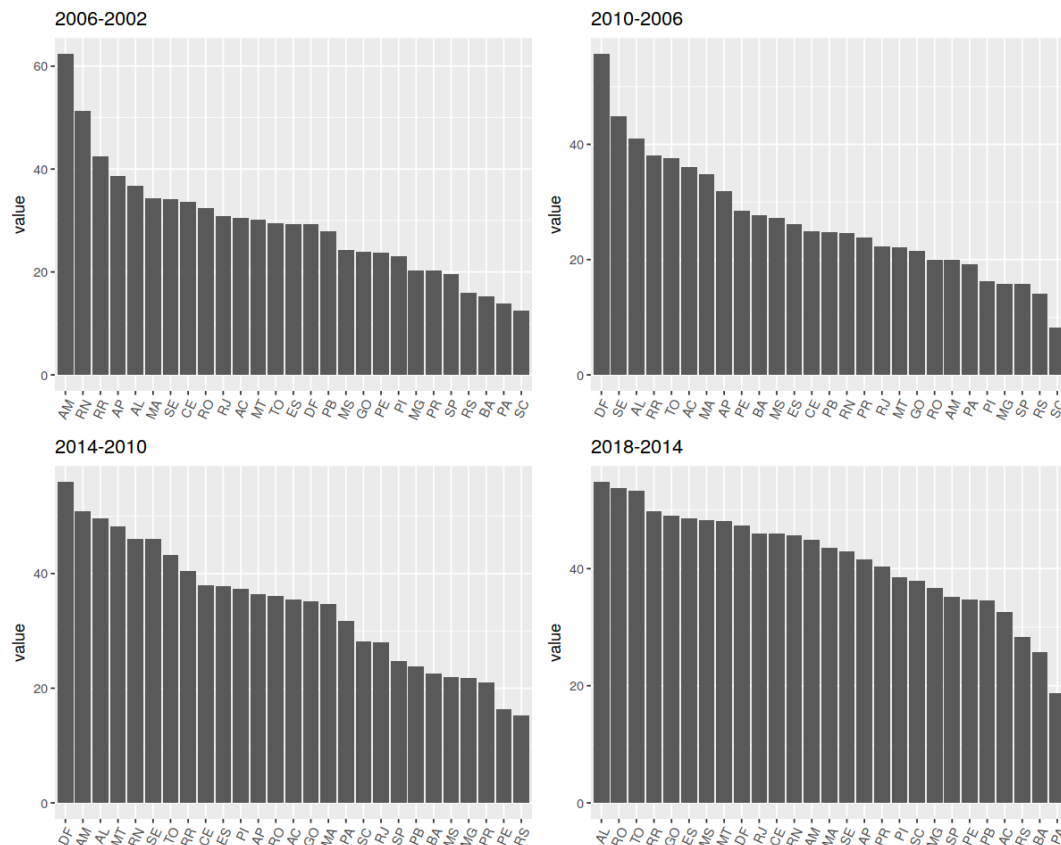
Using SIGLA_UF as id variables

In [12]:

```
p7 <- volatilidade %>% filter(variable == '2002') %>%  
  ggplot(.,aes(reorder(SIGLA_UF, -value), value))+  
  geom_bar(stat = 'identity')+  
  theme(axis.text.x = element_text(angle = 65,hjust=1),  
        axis.title.x = element_blank())+  
  ggtitle('2002-1998')  
  
p8 <- volatilidade %>% filter(variable == '2006') %>%  
  ggplot(.,aes(reorder(SIGLA_UF, -value), value))+  
  geom_bar(stat = 'identity')+  
  theme(axis.text.x = element_text(angle = 65,hjust=1),  
        axis.title.x = element_blank())+  
  ggtitle('2006-2002')  
  
p9 <- volatilidade %>% filter(variable == '2010') %>%  
  ggplot(.,aes(reorder(SIGLA_UF, -value), value))+  
  geom_bar(stat = 'identity')+  
  theme(axis.text.x = element_text(angle = 65,hjust=1),  
        axis.title.x = element_blank())+  
  ggtitle('2010-2006')  
  
p10 <- volatilidade %>% filter(variable == '2014') %>%  
  ggplot(.,aes(reorder(SIGLA_UF, -value), value))+  
  geom_bar(stat = 'identity')+  
  theme(axis.text.x = element_text(angle = 65,hjust=1),  
        axis.title.x = element_blank())+  
  ggtitle('2014-2010')  
  
p11 <- volatilidade %>% filter(variable == '2018') %>%  
  ggplot(.,aes(reorder(SIGLA_UF, -value), value))+  
  geom_bar(stat = 'identity')+  
  theme(axis.text.x = element_text(angle = 65,hjust=1),  
        axis.title.x = element_blank())+  
  ggtitle('2018-2014')
```

In [13]:

```
grid.arrange(p8,p9, p10,p11, ncol=2)
```



Desproporcionalidade

- A desproporcionalidade é a diferença entre os votos e as cadeiras e um partido.
- Usamos aqui fórmula desenvolvida por Loosemore e Hanby (1971).

$$LH = \frac{1}{2} \sum_{i=1}^n (pv_i - pc_i)$$

- Onde:
 - pv é o percentual de votos
 - pc o percentual de cadeiras
- O resultado varia entre 0 e 100.

- Representa o percentual de cadeiras que foram distribuidas desproporcionalmente entre os partidos.

In [14]:

```
base_uf_full <- readRDS('../base_uf_full.Rda')

eleitos <- c('ELEITO', 'ELEITO POR MÉDIA', 'ELEITO POR QP', 'MÉDIA')

n_eleitos_partidos <- base_uf_full %>%
  group_by(ANO_ELEICAO, SIGLA_UF, SIGLA_PARTIDO) %>%
  filter(DESCRICAO_CARGO == 'DEPUTADO FEDERAL', DESC_SIT_CAND_TOT %i
n% eleitos) %>%
  summarise(n_eleitos = n())

n_magnitude <- n_eleitos_partidos %>%
  group_by(ANO_ELEICAO, SIGLA_UF) %>%
  summarise(magnitude = sum(n_eleitos))
```


In [15]:

```
desprop <-  
  merge(  
    base_uf_partido[base_uf_partido$DESCRICA0_CARGO=='DEPUTADO FEDER  
AL',],  
    n_eleitos_partidos,  
    by = c('ANO_ELEICA0', 'SIGLA_UF', 'SIGLA_PARTIDO'),  
    all.x = TRUE  
  )  
  
desprop <- merge(  
  desprop,  
  n_magnitude,  
  by = c('ANO_ELEICA0', 'SIGLA_UF'),  
  all.x = TRUE  
)  
  
desprop$n_eleitos <- ifelse(is.na(desprop$n_eleitos),0,desprop$n)  
desprop$perc_cadeiras <- (desprop$n / desprop$magnitude)*100  
  
df_desprop <-  
desprop %>%  
  filter(PROP_VOTOS > 0) %>%  
  group_by(ANO_ELEICA0, SIGLA_UF) %>%  
  summarise(  
    LH = sum(abs((PROP_VOTOS*100) - perc_cadeiras)) * (1/2)  
  )
```

In [16]:

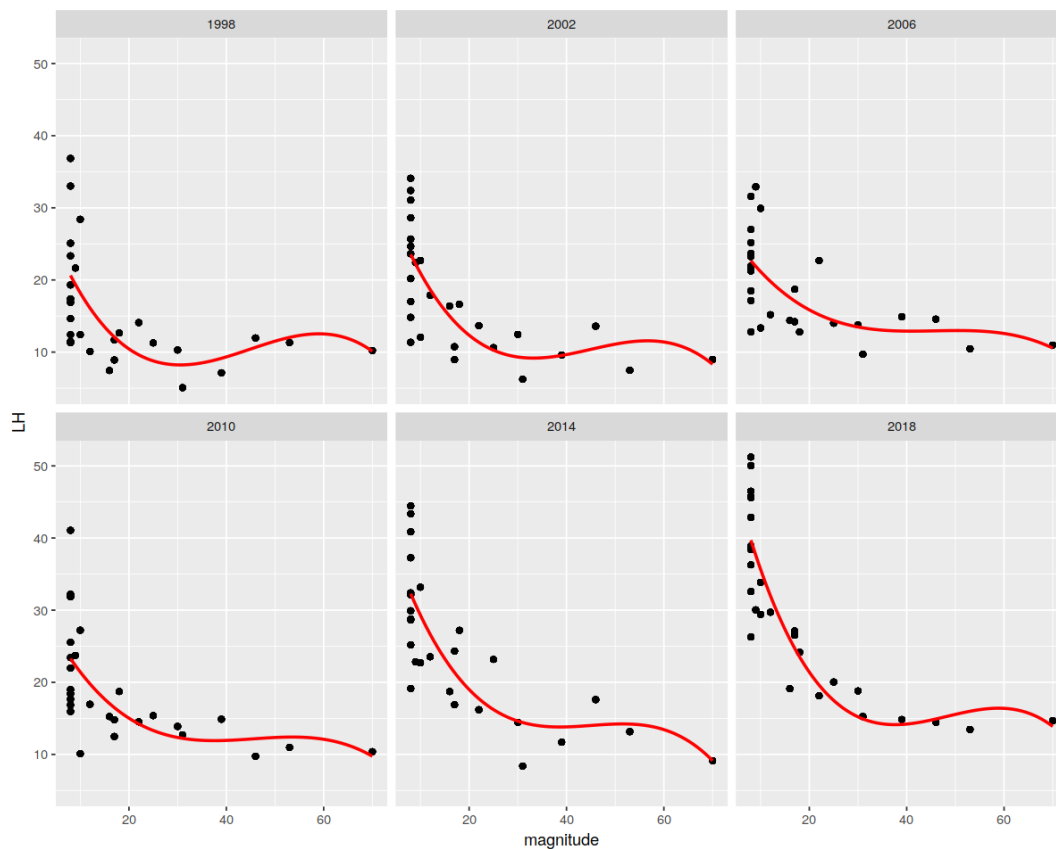
```
df_desprop %>% group_by(ANO_ELEICA0) %>%  
summarise(n = n(), Mínimo = min(LH), Máximo = max(LH),  
          Média = mean(LH), Desvio = sd(LH),  
          CoefVar = Desvio/Média*100)
```

ANO_ELEICA0	n	Mínimo	Máximo	Média	Desv
1998	27	5,069271	36,84941	15,42033	7,936
2002	27	6,233612	34,09645	17,55038	8,038
2006	27	9,697264	32,90992	18,75753	6,593
2010	27	9,753872	41,06395	18,72771	7,590
2014	27	8,390772	44,44481	24,63967	9,973
2018	27	13,464505	51,22837	29,77568	11,99

In [17]:

```
outro_df <-
  merge(
    df_desprop,
    desprop,
    by.x = c('ANO_ELEICA0', 'SIGLA_UF'),
    all.x=TRUE)

ggplot(outro_df, aes(magnitude,LH))+
  geom_point()+
  stat_smooth(method="glm", se=TRUE, fill=NA,
              formula=y ~ poly(x, 3, raw=TRUE),colour="red")+
  facet_wrap(~ANO_ELEICA0)
```

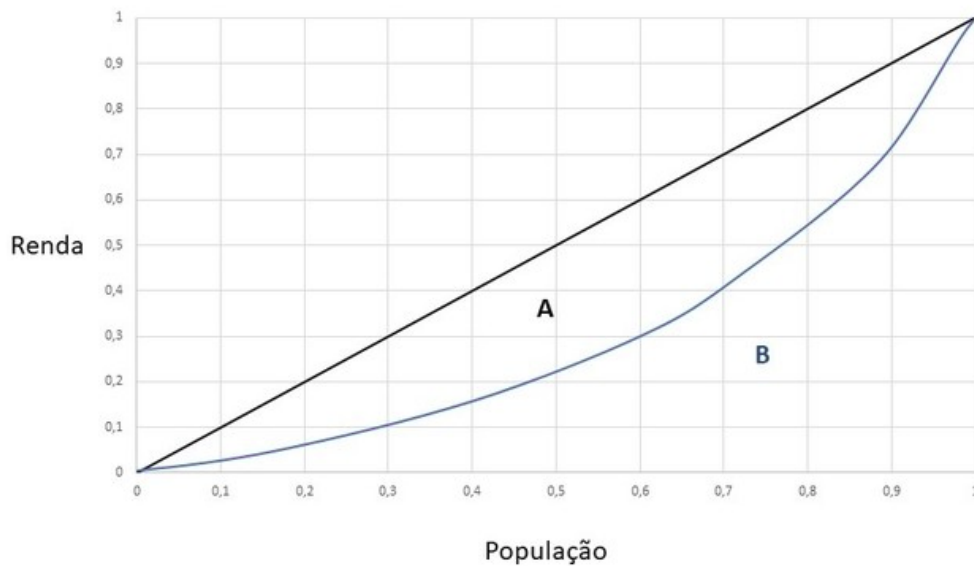


Desigualdade no apoio dos partidos

(nacionalização)

Nacionalização partidária (Gini)

- Desenvolvido por Corrado Gini (1912) para medir a desigualdade, normalmente de renda.
- Como medir?



Representação da Curva de Lorenz

$$G = 1 - \sum_{i=1}^{k=n-1} (P_{k+1} - P_k)(R_{k+1} + R_k)$$

- Onde:
 - P é proporção acumulada da população.
 - R é a proporção acumulada da renda.
- Jones e Mainwaring (2003) aplicaram essa fórmula para calcular a nacionalização dos partidos políticos.
 - A população seria todos os partidos entre os distritos.
 - A renda seria a votação desses partidos nos distritos.
- Quanto mais próximo de 1 mais desigual seria. Jones e Mainwaring subtrai um do G para ter valores altos representando maior nacionalização.

In [18]:

```
nacionalizacao <- base_uf_partido %>%  
  filter(DESCRICA0_CARGO=='DEPUTADO FEDERAL') %>%  
  group_by(ANO_ELEICAO, SIGLA_PARTIDO) %>%  
  arrange(SIGLA_PARTIDO, PROP_VOTOS) %>%  
  summarise( nacionalizacao = 1 - Gini(x = PROP_VOTOS))
```

In [19]:

```
p1 <- ggplot(nacionalizacao[nacionalizacao$ANO_ELEICA0==1998,], aes(
  reorder(SIGLA_PARTIDO, nacionalizacao), nacionalizacao))+
  geom_bar(stat = 'identity')+
  theme(axis.text.x = element_text(angle = 65,hjust=1),
        axis.title.x = element_blank(),
        axis.title.y = element_blank())+ coord_flip()+ggtitle('1998'
)

p2 <- ggplot(nacionalizacao[nacionalizacao$ANO_ELEICA0==2002,], aes(
  reorder(SIGLA_PARTIDO, nacionalizacao), nacionalizacao))+
  geom_bar(stat = 'identity')+
  theme(axis.text.x = element_text(angle = 65,hjust=1),
        axis.title.x = element_blank(),
        axis.title.y = element_blank())+ coord_flip()+ggtitle('2002'
)

p3 <- ggplot(nacionalizacao[nacionalizacao$ANO_ELEICA0==2006,], aes(
  reorder(SIGLA_PARTIDO, nacionalizacao), nacionalizacao))+
  geom_bar(stat = 'identity')+
  theme(axis.text.x = element_text(angle = 65,hjust=1),
        axis.title.x = element_blank(),
        axis.title.y = element_blank())+ coord_flip()+ggtitle('2006'
)

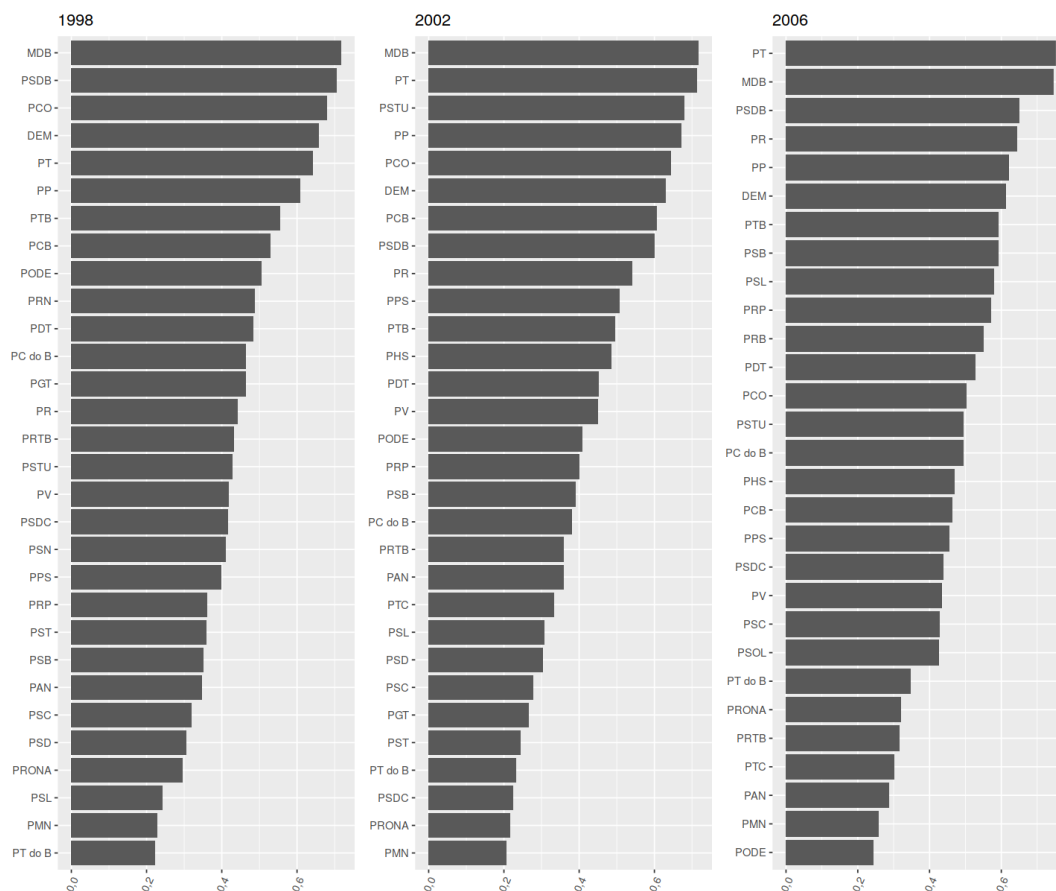
p4 <- ggplot(nacionalizacao[nacionalizacao$ANO_ELEICA0==2010,], aes(
  reorder(SIGLA_PARTIDO, nacionalizacao), nacionalizacao))+
  geom_bar(stat = 'identity')+
  theme(axis.text.x = element_text(angle = 65,hjust=1),
        axis.title.x = element_blank(),
        axis.title.y = element_blank())+ coord_flip()+ggtitle('2010'
)

p5 <- ggplot(nacionalizacao[nacionalizacao$ANO_ELEICA0==2014,], aes(
  reorder(SIGLA_PARTIDO, nacionalizacao), nacionalizacao))+
  geom_bar(stat = 'identity')+
  theme(axis.text.x = element_text(angle = 65,hjust=1),
        axis.title.x = element_blank(),
        axis.title.y = element_blank())+ coord_flip()+ggtitle('2014'
)
```

```
p6 <- ggplot(nacionalizacao[nacionalizacao$ANO_ELEICA0==2018,], aes(
  reorder(SIGLA_PARTIDO, nacionalizacao), nacionalizacao))+
  geom_bar(stat = 'identity')+
  theme(axis.text.x = element_text(angle = 65,hjust=1),
        axis.title.x = element_blank(),
        axis.title.y = element_blank())+ coord_flip()+ggtitle('2018'
  )
```

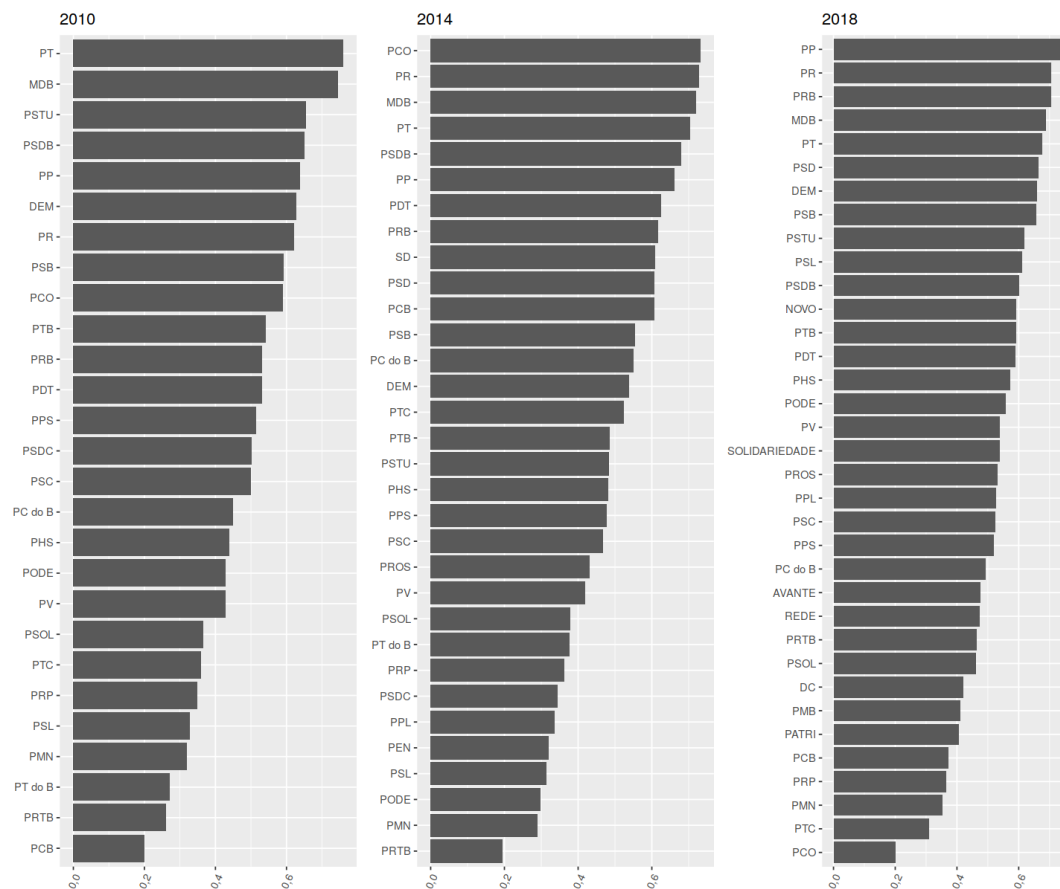
In [20]:

```
options(repr.plot.width=12, repr.plot.height=10)
grid.arrange(p1,p2, p3,ncol=3)
```



In [21]:

```
options(repr.plot.width=12, repr.plot.height=10)
grid.arrange(p4,p5, p6,ncol=3)
```



Ao infinito ...

- Mais medidas
 - Concentração
 - Desequilíbrio
 - Competição
 - Tamanho dos partidos
- Livro

