

# MAD-CB



# Análise de Componentes Principais – Principal Components Analysis (PCA)

# Redução em Número de Variáveis

- Para conjuntos de dados grandes com muitas variáveis
- Necessidade/desejo reduzir dimensões para um número menor coerente
  - ▶ Facilidade de manipulação do conjunto
  - ▶ Coerência de interpretação
- Adjunto útil para técnicas de análise genômicas
  - ▶ NGS, RNASeq, DNaseq
  - ▶ Número enorme de genes, mRNAs, etc.
- Poder de PCA para impôr um ordem aparante num grande corpo de variáveis

- Inventado por Karl Pearson em 1901
- Resumo de padrões de correlações entre variáveis observados
  - ▶ Com objetivo de reduzir o número das variáveis
- Componentes são combinações lineares das variáveis
- Semelhante com Análise de Fatores
  - ▶ *Diferença:*
  - ▶ Análise de Fatores: Quais processos subjacentes teóricos podiam ter criados essas correlações?
  - ▶ PCA: Simples aglomeração das variáveis correlacionadas
- PCA – uma transformação ortogonal de dados correlacionados
  - ▶ **Transformação ortogonal** – uma transformação que leva as variáveis a não ter correlação

# O Que Faz PCA?

- Produz uma matriz de carregamentos (pesos) que são usadas para determinar pontuações (*scores*) para todos os casos combinando as variáveis que compõem o componente

# Quando Usamos PCA?

- Quando queremos ter menos variáveis que colecionou originalmente para fazer análises subsequentes

# Quantos componentes precisamos?

- Depende do ...
  - ① número de variáveis
  - ② grau de correlação entre elas
- Nós instruímos o software sobre o número de componentes
  - ▶ Baseado num estudo preliminar e o “screeplot”
- Com dados altamente correlacionados, precisa menos componentes para enquadrar a variância no modelo
- Com dados menos correlacionados, precisa mais
- Procedimento de testar para um número “ideal” dos componentes

- O algoritmo calcula o número desejado de componentes
- Começa com o componente que maximiza a variância explicada (PC1)
  - ▶ Por causa que minimiza a soma dos quadrados das distâncias entre os pontos e a linha que o componente descreve
- 2º componente é outra combinação linear que maximiza variância
  - ▶ **MAS**, a direção dele é perpendicular ao PC1, que quer dizer ...
  - ▶ é ortogonal a PC1 (rotação dos eixos do componente)
  - ▶ elimina a correlação incluído em PC1



# Um Pouco Dentro de Caixa Preta de PCA

- PCA funciona como aplicação da técnica de álgebra linear - SVD
- Decomposição em Valores Singulares
  - ▶ *Singular Value Decomposition*
  - ▶ Fatoração de uma matriz de números reais
- Uma matriz  $X$  pode ser fatorada em 3 submatrizes:

$$X = UDV^T$$

- Se os dados foram normalizados, a matriz  $T$  têm os componentes principais
- $T$  é uma matriz ortogonal

# Matriz Ortogonal – 2 Características

- ① Matriz sempre pode ser invertida [ $X^{-1} = X^T$ ]
- ② Produto da matriz e sua transposta é a matriz de identidade [ $XX^T = I$ ]

# Matriz de Identidade (I)

- Matriz quadrada com 1's nos diagonais e 0's nas outras posições
- Matriz de Identidade  $3 \times 3$ :

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

# Requisitos de Dados

- Dados devem ser numéricos
  - ▶ Variáveis categóricas não podem ser analisadas
- Normalidade
  - ▶ Não precisam ser normais
  - ▶ Se foram normais, resultado vai explicar mais da variância
- Outliers
  - ▶ Outliers podem exercer uma influência exagerada sobre o resultado
- Linearidade
  - ▶ Componentes são combinações lineares das variáveis
  - ▶ Baseado em correlação que é linear em si

- Antigamente, a “regra de ouro” foi muitos casos por cada variável no PCA
  - ▶ Pelo menos 5 por variável
- Hoje estamos mais agnóstico sobre a quantidade; é a qualidade que conta
- Mas, mais é melhor
- Amostras pequenas estão menos estáveis

## 2 Algoritmos entre Muitos

- principal do pacote psych
- prcomp do base R
- Outros programas
  - ▶ corrplot – visualização de matriz de correlação
  - ▶ screeplot – visualização do impacto de cada componente (2 versões)
  - ▶ biplot de pacote psych – visualização de carregamentos dos 1º dois componentes

# Mergulhar Num Exemplo

- Dados de Câncer de Mama de Wisconsin de novo
- Características dos tumores de mama
- Variável dependente: diagnose (diag)
  - ▶ Que não nos interesse aqui
  - ▶ Só queremos estudar os covariáveis

# Carregar Dados e Eliminar NA's

- 699 casos de 9 (co)variáveis
- Em `bare_nuclei`, valor imputado no lugar de “?” será a média da variável



# Explorar a Matriz de Correlações

- Construa uma matriz (formato de tibble) de correlações entre as covariáveis
- Faça um gráfico que mostra elas

```
bccorr <- as_tibble(cor(bcmmod, method = "pearson"))  
  
meancors <- bccorr %>% map_dbl(~ mean(., na.rm = TRUE))  
meancorsp <- rownames_to_column(as.data.frame(meancors)) %>%  
  arrange(desc(abs(meancors)))
```

# Matriz de Correlação

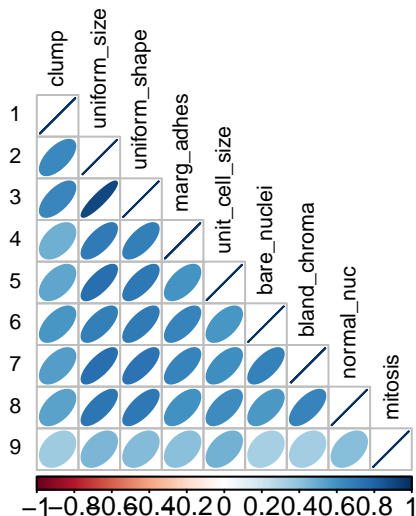
```
## # A tibble: 9 × 9
##       clump uniform_size uniform_shape marg_adhes unit_cell_size
##       <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1 1.0000000    0.6449125    0.6545891    0.4863562    0.5218162
## 2 0.6449125    1.0000000    0.9068819    0.7055818    0.7517991
## 3 0.6545891    0.9068819    1.0000000    0.6830792    0.7196684
## 4 0.4863562    0.7055818    0.6830792    1.0000000    0.5995991
## 5 0.5218162    0.7517991    0.7196684    0.5995991    1.0000000
## 6 0.5872997    0.6868014    0.7096058    0.6650492    0.5812613
## 7 0.5584282    0.7557210    0.7359485    0.6667153    0.6161018
## 8 0.5358345    0.7228648    0.7194463    0.6033524    0.6288807
## 9 0.3500339    0.4586931    0.4389109    0.4176328    0.4791015
## # ... with 4 more variables: bare_nuclei <dbl>, bland_chroma <dbl>,
## #   normal_nuc <dbl>, mitosis <dbl>
```

# Média de Correlações

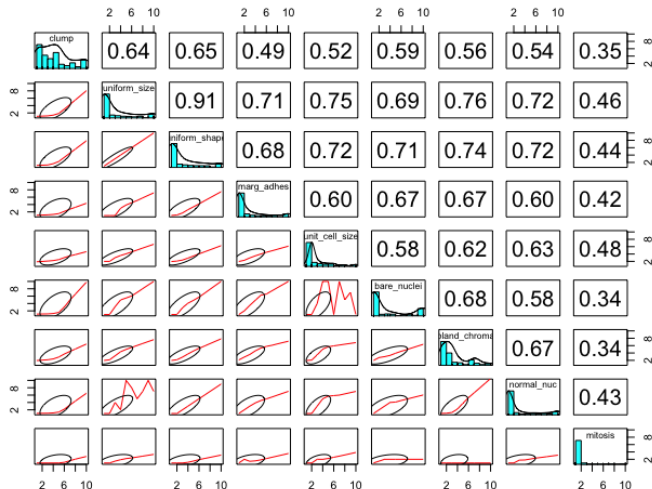
##	rowname	meancors
## 1	uniform_size	0.7370284
## 2	uniform_shape	0.7297922
## 3	bland_chroma	0.6687620
## 4	unit_cell_size	0.6553587
## 5	normal_nuc	0.6535505
## 6	marg_adhes	0.6474851
## 7	bare_nuclei	0.6468905
## 8	clump	0.5932523
## 9	mitosis	0.4728464

# Gráfico de Correlações

```
corrplot::corrplot(as.matrix(bccorr), method = "ellipse", type = "lower",  
  tl.cex = .7, tl.col = "black")
```



# Gráfico Alternativo – psych::pairs.panels



# Construir PCA Inicial – Para Testar # de Componentes

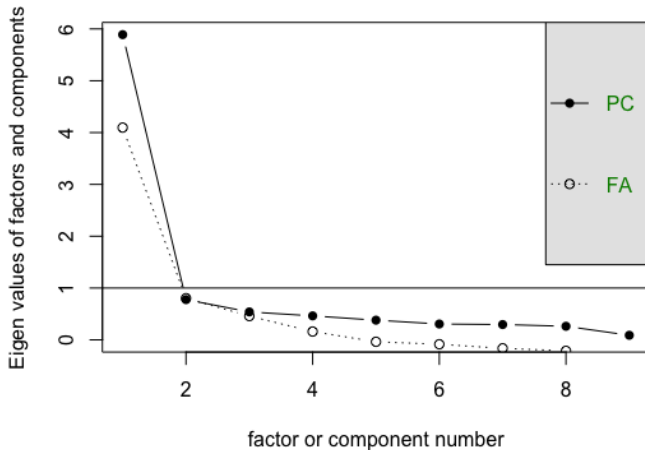
```
set.seed(42)
bcpca1 <- principal(bcmmod, nfactors = 8, rotate = "none", missing = TRUE)
```

- `nfactors = 8` – máximo possível; = 1 a menos do número de variáveis

# Screeplot do Primeiro Modelo – do Pacote psych

```
scree(bccorr, main = "Scree Plot de principal")
```

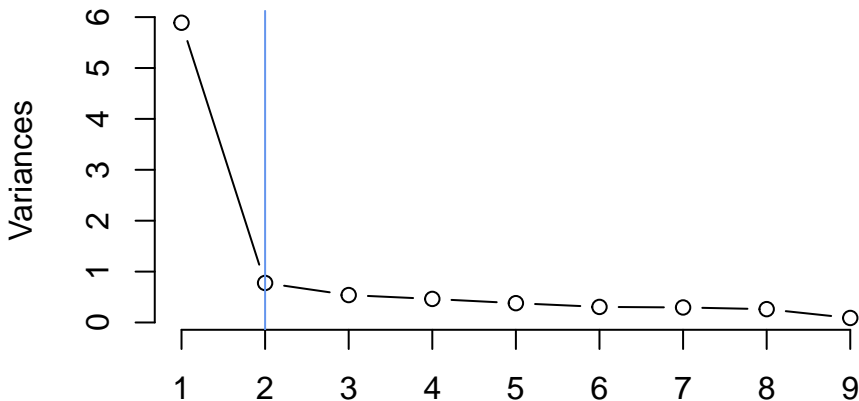
Scree Plot de principal



## Screeplot do Primeiro Modelo – do base R

```
screeplot(prcomp(bcmmod, rank = 8, center = TRUE,  
                scale = TRUE), type = "l", main = "Scree Plot de base")  
abline(v = 2, col = "cornflowerblue")
```

### Scree Plot de base





# Como Julgar o Número de Componentes do Modelo Inicial no “Scree plot”

- Primeiro número de componente com “Eigen value” abaixo de 1
  - ▶ scree do pacote psych
- Primeiro número de componente depois de “cotovelo” no gráfico
  - ▶ screeplot do base R
- Conclusão
  - ▶ Pode trabalhar com 2 componentes

# 1º Modelo em Números

```
bcpca1$loadings
```

```
##
## Loadings:
##          PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8
## clump      0.735 -0.126  0.633          -0.152          0.119
## uniform_size 0.925          -0.139          -0.108 -0.231
## uniform_shape 0.917          -0.117          -0.302
## marg_adhes  0.808          -0.312  0.316          -0.376
## unit_cell_size 0.816  0.146          -0.267 -0.413          0.233
## bare_nuclei 0.809 -0.224          0.364          0.334  0.209
## bland_chroma 0.840 -0.202 -0.143          0.155  0.140 -0.383  0.206
## normal_nuc  0.815          -0.304  0.400          0.264
## mitosis     0.557  0.799          0.169
##
##          PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8
## SS loadings  5.890 0.777 0.539 0.463 0.380 0.306 0.295 0.262
## Proportion Var 0.654 0.086 0.060 0.051 0.042 0.034 0.033 0.029
## Cumulative Var 0.654 0.741 0.801 0.852 0.894 0.928 0.961 0.990
```

# Construir Modelo com 2 Componentes

```
set.seed(42)
bcpca2 <- principal(bcmmod, nfactors = 2, rotate = "none",
                    missing = TRUE)
```

# Resumo do Modelo

```
summary(bcpca2)
```

```
##  
## Factor analysis with Call: principal(r = bcpca2, nfactors = 2, rotate = "none", missing = TRUE)  
##  
## Test of the hypothesis that 2 factors are sufficient.  
## The degrees of freedom for the model is 19 and the objective function was 0.5  
## The number of observations was 699 with Chi Square = 346.31 with prob < 5.9e-62  
##  
## The root mean square of the residuals (RMSA) is 0.05
```

# Carregamentos das Variáveis sobre os Componentes – bcpca2

```
bcpca2$loadings
```

```
##  
## Loadings:  
##          PC1    PC2  
## clump      0.735 -0.126  
## uniform_size 0.925  
## uniform_shape 0.917  
## marg_adhes  0.808  
## unit_cell_size 0.816  0.146  
## bare_nuclei 0.809 -0.224  
## bland_chroma 0.840 -0.202  
## normal_nuc  0.815  
## mitosis     0.557  0.799  
##  
##          PC1    PC2  
## SS loadings  5.890 0.777  
## Proportion Var 0.654 0.086  
## Cumulative Var 0.654 0.741
```

# Modelo Suficiente Bom?

- Descreve 74.1% da variância
  - ▶ OK, mas não muito bom
- Se aumentamos até 3 componentes Pode aumentar proporção da variância descrita até 80%
- **Porém** vou continuar com 2 componentes para facilitar a apresentação

# Construir Modelo Final - 2 Componentes com Rotação

- Agora, testar o modelo sem e com a rotação varimax
  - ▶ varimax: rotação que maximiza a diferença entre os componentes
  - ▶ Facilita interpretação dos componentes
- Conseguimos reduzir o número de variáveis de 9 até 2 (ou 3)

## Modelo 2 – com Rotatção

```
set.seed(42)
bcpca2r <- principal(bcmmod, nfactors = 2, rotate = "varimax",
                     missing = TRUE)
```



# Modelo 3 – sem Rotatção – Carregamentos

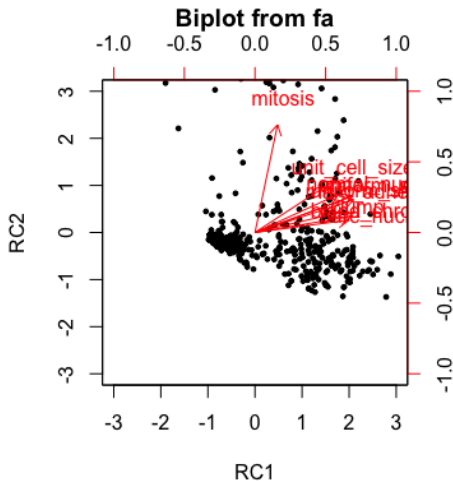
```
bcpca2r$loadings
```

```
##  
## Loadings:  
##          RC1    RC2  
## clump      0.725 0.172  
## uniform_size 0.868 0.323  
## uniform_shape 0.873 0.290  
## marg_adhes  0.758 0.281  
## unit_cell_size 0.694 0.454  
## bare_nuclei 0.832 0.110  
## bland_chroma 0.852 0.143  
## normal_nuc  0.742 0.339  
## mitosis     0.200 0.954  
##  
##          RC1    RC2  
## SS loadings  5.106 1.560  
## Proportion Var 0.567 0.173  
## Cumulative Var 0.567 0.741
```

- Eixos a esquerda, fundo (*ticks* em preto)
  - ▶ Pontuações no cada componente para os casos
- Eixos a direita, acima (*ticks* em vermelho)
  - ▶ Carregamentos das variáveis

# Biplot dos Componentes

```
biplot(bcpca2r, main = "Biplot -- Câncer de Mama -- v.2r", pch = 19)
```



# Como Usamos os Resultados da PCA?

- Sistema calcula pontuação para cada caso (paciente) para cada componente

$$P_{\text{caso},pc} = L_{v1,pc} * V_{v1,pc} + L_{v2,pc} * V_{v2,pc} + \dots + L_{vn,pc} * V_{vn,pc}$$

- onde  $P$  = pontuação,  $pc$  = componente,  $L$  = Carregamento,  $v1...vn$  = variáveis,  $V$  = valor da variável
- Cada caso terá uma pontuação para cada componente
  - ▶ Como o componente funcionou como uma variável
- Carregado no elemento `$scores` do modelo (neste caso, `bcpca2r`)
- **Precisa lembrar que os valores da PCA são normalizados**

# Pode Pôr as Pontuações na Estrutura de Dados para Usar nas Análises Subsequentes

- Mais, restorar os identificadores ao tibble

```
bcid <- as_tibble(bc$sample_code_number)
colnames(bcid) <- 'id'
compscores <- as_tibble(bcpca2r$scores)
bcpdados <- bind_cols(bcid, bcpmod, compscores)
glimpse(bcpdados)
```

```
## Observations: 699
## Variables: 12
## $ id          <int> 1000025, 1002945, 1015425, 1016277, 1017023, 10...
## $ clump       <dbl> 5, 5, 3, 6, 4, 8, 1, 2, 2, 4, 1, 2, 5, 1, 8, 7,...
## $ uniform_size <dbl> 1, 4, 1, 8, 1, 10, 1, 1, 1, 2, 1, 1, 3, 1, 7, 4...
## $ uniform_shape <dbl> 1, 4, 1, 8, 1, 10, 1, 2, 1, 1, 1, 1, 3, 1, 5, 6...
## $ marg_adhes   <dbl> 1, 5, 1, 1, 3, 8, 1, 1, 1, 1, 1, 1, 3, 1, 10, 4...
## $ unit_cell_size <dbl> 2, 7, 2, 3, 2, 7, 2, 2, 2, 2, 2, 1, 2, 2, 2, 7, 6...
## $ bare_nuclei   <dbl> 1.000000, 10.000000, 2.000000, 4.000000, 1.0000...
## $ bland_chroma  <dbl> 3, 3, 3, 3, 3, 9, 3, 3, 1, 2, 3, 2, 4, 3, 5, 4,...
## $ normal_nuc    <dbl> 1, 2, 1, 7, 1, 7, 1, 1, 1, 1, 1, 1, 4, 1, 5, 3,...
## $ mitosis       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1, 4, 1,...
## $ RC1           <dbl> -0.5090005, 0.7988633, -0.5689635, 0.8134106, -...
## $ RC2           <dbl> -0.34231373, -0.33805006, -0.33023349, -0.33338...
```

## Exemplo 2

- Medidas de habilidades cognitivas numa amostra de pacientes
- 74 testes psicológicos administrados aos pacientes
- Para fazer parte de uma análise ANOVA com outras variáveis não-neurocientíficas
- Com PCA, reduzimos os 74 testes em 6 componentes.

# Carregar os Dados

- Dados já reduzidos aos numéricas que mostram valores “raw” dos testes (33 vars + id)

```
load("excog.RData")
glimpse(cogtest)
```

```
## Observations: 21
## Variables: 34
## $ codigo          <fctr> SAA25, SAA34, SAA38, SAA24, SAA29, SAA04, ...
## $ srq             <dbl> 1, 1, 1, 2, 2, 3, 3, 3, 6, 6, 7, 7, 7, 8, 1...
## $ time_sec_tmt_a   <dbl> 43, 34, 55, 38, 52, 71, 72, 37, 42, 53, 15,...
## $ time_sec_tmt_b   <dbl> 144, 66, 207, 254, 409, 275, 290, 74, 135, ...
## $ time_stroop_a     <dbl> 16, 22, 17, 22, 14, 15, 34, 13, 21, 17, 11,...
## $ time_stroop_b     <dbl> 22, 23, 43, 26, 17, 20, 35, 18, 22, 27, 12,...
## $ time_stroop_c     <dbl> 31, 35, 51, 63, 40, 30, 49, 22, 44, 33, 20,...
## $ total_fig_rey_imed <dbl> 31.0, 27.0, 26.0, 26.0, 28.0, 34.5, 23.5, 3...
## $ total_fig_rey_mem <dbl> 17.5, 10.5, 9.0, 13.0, 9.5, 19.0, 7.0, 20.5...
## $ dod_total        <dbl> 7, 8, 8, 4, 6, 6, 6, 4, 10, 4, 11, 8, 5, 8,...
## $ doi_total        <dbl> 7, 3, 4, 4, 3, 3, 3, 5, 6, 4, 5, 7, 4, 3, 5...
## $ digit_total      <dbl> 14, 11, 12, 8, 9, 9, 9, 9, 16, 8, 16, 15, 9...
## $ cod_total        <dbl> 30, 38, 19, 44, 32, 38, 26, 56, 29, 40, 64,...
## $ proc_simb_total   <dbl> 15, 19, 11, 17, 14, 21, 17, 28, 18, 18, 39,...
## $ ivp              <dbl> 92, 100, 97, 108, 97, 116, 105, 124, 100, 1...
## $ vocab_total       <dbl> 22, 17, 36, 32, 27, 44, 27, 38, 35, 31, 68,...
## $ cubos_total       <dbl> 14, 10, 6, 6, 6, 6, 16, 3, 20, 22, 12, 47, 52,...
## $ semelrh_total     <dbl> 14, 16, 16, 19, 26, 22, 16, 35, 22, 18, 35,...
## $ qi_verbal         <dbl> 64, 63, 81, 78, 80, 91, 74, 95, 82, 74, 116...
## $ qi_exec           <dbl> 75, 63, 83, 66, 67, 97, 73, 92, 89, 66, 120...
## $ ravlt_ai         <dbl> 6, 6, 4, 6, 7, 3, 6, 7, 4, 3, 9, 6, 2, 3, 5...
## $ ravlt_a2         <dbl> 10, 8, 6, 10, 10, 5, 8, 13, 6, 3, 13, 9, 5,
```



# Examinar Correlações entre as Variáveis

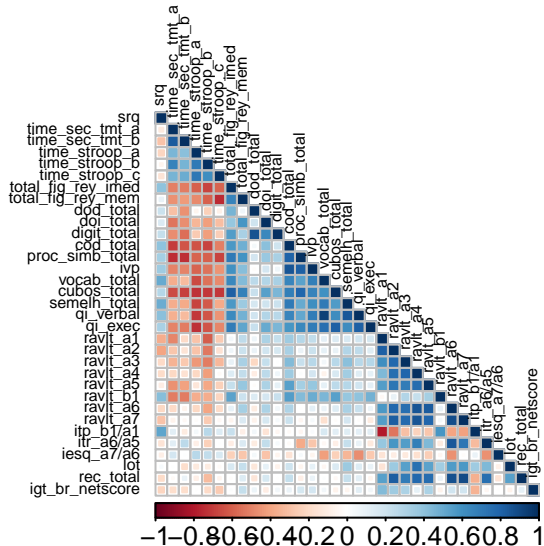
```
## use non-parametric correlation
corcogtest <- as_tibble(cor(cogtest[,-1], method = "spearman"))
meancors <- corcogtest %>% map_dbl(~ mean(., na.rm = TRUE))
meancorsp <- as_tibble(rownames_to_column(as.data.frame(meancors))) %>%
  arrange(desc(abs(meancors)))
```

# Tabela das Médias das Correlações entre as Variáveis

```
kable(meancorsp)
```

rowname	meancors
ravlt_a5	0.2668321
cod_total	0.2527272
ravlt_a4	0.2518546
semelh_total	0.2467428
ravlt_a6	0.2465416
ravlt_a3	0.2434324
qi_verbal	0.2341593
ravlt_a2	0.2341515
time_stroop_b	-0.2325217
rec_total	0.2271888
ravlt_a7	0.2230976
ivp	0.2148754
total_fig_rey_imed	0.2147870
qi_exec	0.2094449
cubos_total	0.2045991
lot	0.2020018
total_fig_rey_mem	0.1990659
vocab_total	0.1943698
ravlt_b1	0.1923398
time_stroop_a	-0.1764982
proc_simb_total	0.1752442
doi_total	0.1748589
time_sec_tmt_a	-0.1651310
time_sec_tmt_b	-0.1644411
ravlt_a1	0.1583766
itr_a6/a5	0.1540949
digit_total	0.1468896
time_stroop_c	-0.1329364

```
corrplot(cor(cogtest[,-1], method = "spearman"),
  method = "square", type = "lower",
  tl.cex = .5, tl.col = "black")
```



# Resultados de Correlação

- Fora de alguns grupos pequenos de variáveis, elas não têm correlações muito altas
- Com correlações mais baixas, pode esperar a necessidade de mais componentes
- NB o uso de quadrados invés de elipses – espaços muito menores
- Usar algoritmo `prcomp` de base R para fazer os calculos
  - ▶ Precisamos especificar a normalização dos dados
  - ▶ `center = TRUE, scale = TRUE`

# Construir um Modelo Inicial para Testar o Número de PCs

```
pcacalc <- cogtest[,2:ncol(cogtest)] ## criar dados sem o id (factor, não número)
set.seed(42)
pcafit1 <- prcomp(pcacalc, rank = 10, scale = TRUE, center = TRUE)
```

# Resumo do Modelo

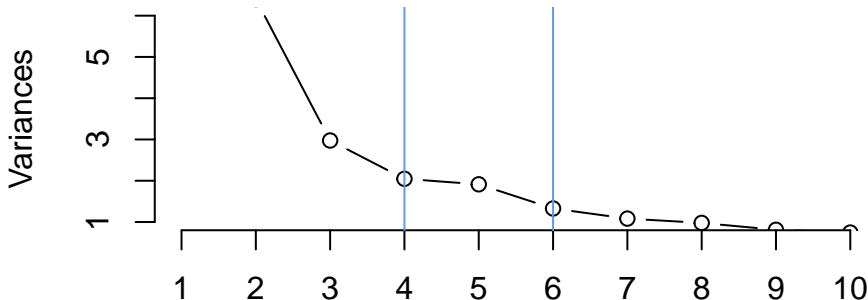
```
summary(pcafit1)
```

```
## Importance of first k=10 (out of 21) components:
##
##          PC1      PC2      PC3      PC4      PC5      PC6
## Standard deviation    3.5106 2.5304 1.72474 1.43068 1.38330 1.15220
## Proportion of Variance 0.3735 0.1940 0.09014 0.06203 0.05799 0.04023
## Cumulative Proportion 0.3735 0.5675 0.65764 0.71967 0.77765 0.81788
##
##          PC7      PC8      PC9      PC10
## Standard deviation    1.04030 0.98876 0.90057 0.86275
## Proportion of Variance 0.03279 0.02963 0.02458 0.02256
## Cumulative Proportion 0.85068 0.88030 0.90488 0.92744
```

# Screeplot

```
screeplot(pcafit1, type = "l", ylim = c(1,6), main = "Screeplot - pcafit1")  
abline(v = 4, col = "cornflowerblue")  
abline(v = 6, col = "cornflowerblue")
```

## Screeplot – pcafit1



# Quantos Componentes?

- Cotovelos ocorrem aos **4** e aos **6** componentes (linhas azuis)
- Pode usar um ou outro
- Porém, por causa de baixo número de casos (21), prefiro usar 6 componentes
- 6 toma em conta 82% da variância invés de 72%



# Construir Modelo Final

```
set.seed(42)
pcafit2 <- prcomp(pcacalc, rank = 6, scale = TRUE,
                  center = TRUE, retx = TRUE)
pcaimportance <- tidy(pcafit2, matrix = "d") %>%
  slice(1:6)
kable(pcaimportance)
```

PC	std.dev	percent	cumulative
1	3.510645	0.37347	0.37347
2	2.530382	0.19403	0.56750
3	1.724741	0.09014	0.65764
4	1.430684	0.06203	0.71967
5	1.383305	0.05799	0.77765
6	1.152200	0.04023	0.81788

# Criar Pontuação e Carregamentos

```
cogpca <- as.tibble(pcafit2$x) %>%  
  add_column(codigo = cogtest$codigo, .before = "PC1")  
cogloads <- as.tibble(pcafit2$rotation) %>%  
  add_column(variavel = colnames(cogtest[,-1]), .before = "PC1")
```

# PC Pontuação para os Pacientes

```
kable(cogpca)
```

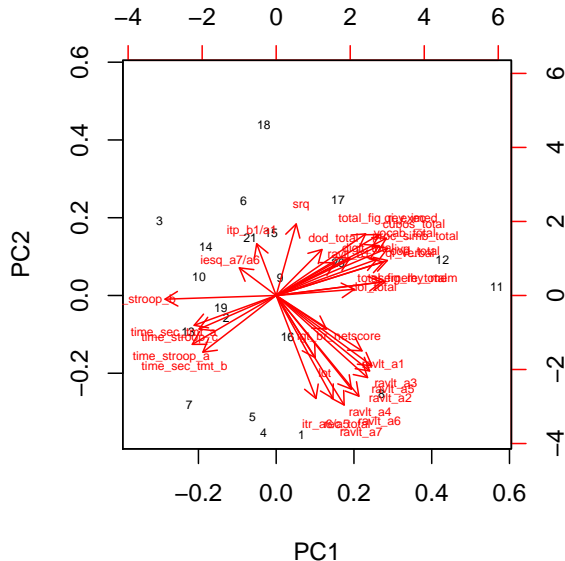
codigo	PC1	PC2	PC3	PC4	PC5	PC6
SAA25	1.0544568	-4.1455409	0.6569173	1.2861027	-2.5786033	-1.1358213
SAA34	-2.0661819	-0.6707727	2.1637158	2.7903071	0.9041903	-0.2493921
SAA38	-4.8327795	2.2266996	2.2870544	-1.0530445	-0.4387162	1.4519428
SAA24	-0.5270424	-4.1073200	-0.9984134	1.2285155	1.1415258	-0.5131524
SAA29	-0.9813072	-3.6122296	0.8149807	-1.8370149	0.7498663	-0.2419661
SAA04	-1.3506479	2.8322995	0.0211974	-1.0880352	2.1874490	0.7588781
SAA18	-3.5969929	-3.2584770	1.8289886	0.0411406	0.9686787	1.3360876
SAA33	4.3613602	-2.9426444	0.3856916	-0.4052328	2.0378401	-1.2125381
SAA05	0.1616059	0.5154849	0.4111294	0.0396129	-3.6007890	0.1782160
SAA20	-3.1912523	0.5572225	0.0443924	-1.9582898	0.0388574	-2.4440400
SAA31	9.1355470	0.2522344	1.7929648	0.2731236	0.5114012	2.0914053
SAA35	6.8780283	1.0806992	-0.2831348	-1.8709003	-1.3530647	-0.6438066
SAA36	-3.6384217	-1.0934636	-2.4734433	-0.4496959	-0.6415100	-0.2121997
SAA28	-2.9114709	1.4461164	0.2086316	1.0449864	-0.2287983	0.2112906
SAA10	-0.2106229	1.8587515	0.5896946	-0.6805372	0.3486266	-0.7854365
SAA21	0.4716043	-1.2312153	-4.7698613	-0.1696474	0.7759010	1.1917531
SAA26	2.5581398	2.8573070	1.7534238	0.5748899	0.4420199	-1.3109261
SAA37	-0.5119386	5.0935621	-1.2696446	0.7537838	0.7799330	-1.1350858
SAA15	-2.2888551	-0.3598593	0.1393074	-2.1174212	-1.1757901	1.5331226
SAA32	2.5632073	0.9669637	-2.3954186	0.5527506	-0.2623551	0.6647815
SAA14	-1.0764362	1.7341821	-0.9081738	3.0446061	-0.6066626	0.4668870

# PC Carregamento para 1º 10 Variáveis

```
kable(cogloads[1:10,])
```

variavel	PC1	PC2	PC3	PC4	PC5	PC6
srq	0.0420233	0.2082761	-0.2695919	0.0947505	-0.1147054	0.0826684
time_sec_tmt_a	-0.1722165	-0.0869178	-0.2132684	-0.2124635	0.0609703	0.2764389
time_sec_tmt_b	-0.1538814	-0.1650981	-0.0063179	-0.3665132	0.0996178	0.1491159
time_stroop_a	-0.1761985	-0.1423664	0.1148217	0.1795451	-0.0302549	0.1911785
time_stroop_b	-0.2334415	-0.0107946	-0.0169558	0.0524373	-0.0558582	0.1894170
time_stroop_c	-0.1619489	-0.0976871	-0.0407716	0.1838451	-0.0488584	0.1696254
total_fig_rey_imed	0.1877155	0.1794291	-0.0405482	0.0451687	-0.0912177	0.0096453
total_fig_rey_mem	0.2206826	0.0388711	0.0711343	-0.0427502	-0.0557209	-0.0417873
dod_total	0.0961479	0.1338473	0.1538545	0.1602330	-0.3570586	0.4770860
doi_total	0.1639769	0.0179340	0.0922253	0.0176002	-0.4311252	-0.2949030

## Biplot de PC1 e PC2



# Trabalho do Pesquisador Agora É Dar Nomes a Esses Componentes

- Examinar quais têm o mais positivos e negativos carregamentos
- PC tem relação conceitual com os positivos e é negativo para os outras.

# Exemplo – PC1 – Carregamentos Altos

```
cogloads %>% group_by(variavel) %>% tally(PC1) %>% top_n(5)
```

```
## Selecting by n
```

```
## # A tibble: 5 × 2
```

```
##       variavel      n
##       <chr>      <dbl>
## 1     cod_total 0.2334268
## 2    cubos_total 0.2313506
## 3 proc_simb_total 0.2339691
## 4     qi_verbal 0.2248814
## 5    semelh_total 0.2292407
```

# Exemplo – PC1 – Carregamentos Mais Negativos

```
cogloads %>% group_by(variavel) %>% tally(PC1) %>% top_n(-5)
```

```
## Selecting by n
```

```
## # A tibble: 5 × 2
```

```
##       variavel      n
##       <chr>      <dbl>
## 1 time_sec_tmt_a -0.1722165
## 2 time_sec_tmt_b -0.1538814
## 3 time_stroop_a  -0.1761985
## 4 time_stroop_b -0.2334415
## 5 time_stroop_c -0.1619489
```



- Empirical Bayes
- Dia para submissão dos relatórios