# Session 09:
# Probleme Linearer Regression

Dominic Schmitz & Janina Esser

Verein für Diversität in der Linguistik

# Kollinearität und andere Probleme

- Bisher haben wir relativ naiv Modelle erstellt
  - irgendeine Variable wird vorhergesagt
  - irgendwelche Variablen sagen vorher
- Problem: Potentielle Gefahren
  1. Nicht-Normalverteilung der Variablen
  2. Kollinearität
  3. Interaktionen
  4. Geringe Datenmenge / zu weniger „Power“

# Kollinearität und andere Probleme

- Bisher haben wir relativ naiv Modelle erstellt
  - irgendeine Variable wird vorhergesagt
  - irgendwelche Variablen sagen vorher
- Problem: Potentielle Gefahren
  1. Nicht-Normalverteilung der Variablen
  2. Kollinearität
  3. Interaktionen
  4. Geringe Datenmenge / zu weniger „Power“

# Kollinearität und andere Probleme

- Bisher haben wir relativ naiv Modelle erstellt
  - irgendeine Variable wird vorhergesagt
  - irgendwelche Variablen sagen vorher
- Problem: Potentielle Gefahren
  1. Nicht-Normalverteilung der Variablen ✓
  2. Kollinearität
  3. Interaktionen
  4. Geringe Datenmenge / zu weniger „Power“

# Kollinearität

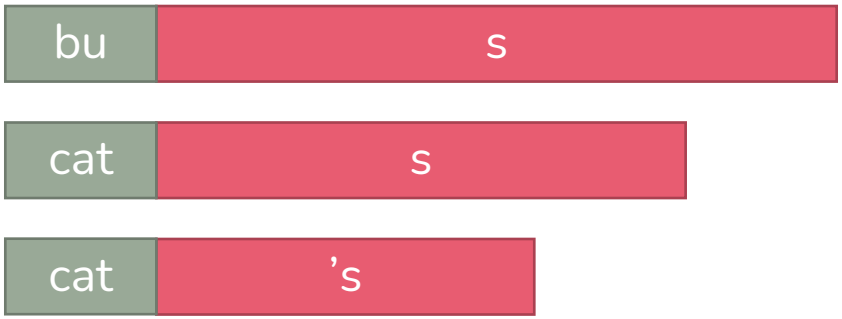
- sind Variablen stark miteinander korreliert, liegt potenziell das Problem der Kollinearität vor
- bedenkt man dieses nicht, so können
  - Koeffizienten nicht mehr zuverlässig berechnet werden
  - werden womöglich Effekte (nicht) signifikant, die eigentlich (nicht) signifikant sind (Stichwort *overfitting*)
- zur Vermeidung von Kollinearität gibt es einige Methoden (siehe z.B. Tomaschek et al., 2018)
- wir schauen uns zunächst die einfachste dieser Methoden genauer an

# Kollinearität

- Für die folgenden Beispiele werden wir Daten folgender Studie nutzen:

**The duration of word-final /s/ differs across morphological categories in English: Evidence from pseudowords**[1]

- Wort-finales /s/ zeigt je nach Morphologie unterschiedliche Dauern




[1] Schmitz, D., Baer-Henney, D., & Plag, I. (2021). The duration of word-final /s/ differs across morphological categories in English: Evidence from pseudowords. Phonetica, 78(5-6), 571-616. doi: 10.1515/phon-2021-2013

# Kollinearität

- Da wir mittlerweile wissen, dass man viele Variablen in ein volles Modell einbauen kann, schauen wir uns auch entsprechend viele Variablen an, nämlich:

  ITEM, SPEAKER, GENDER, TYPEOFS, FOLSEG, SLIDENUMBER, TRANSCRIPTION, BASEDURLOG, BIPHONEPROBSUM, BIPHONEPROBSUMBIN, AGE, LOCATION, MONOMULTILINGUAL, SPEAKINGRATE, PAUSEDUR, PAUSEBIN, FOLTYPE, PREC

- Wir überprüfen jede dieser Variablen hinsichtlich ihrer Korrelation mit allen anderen Variablen

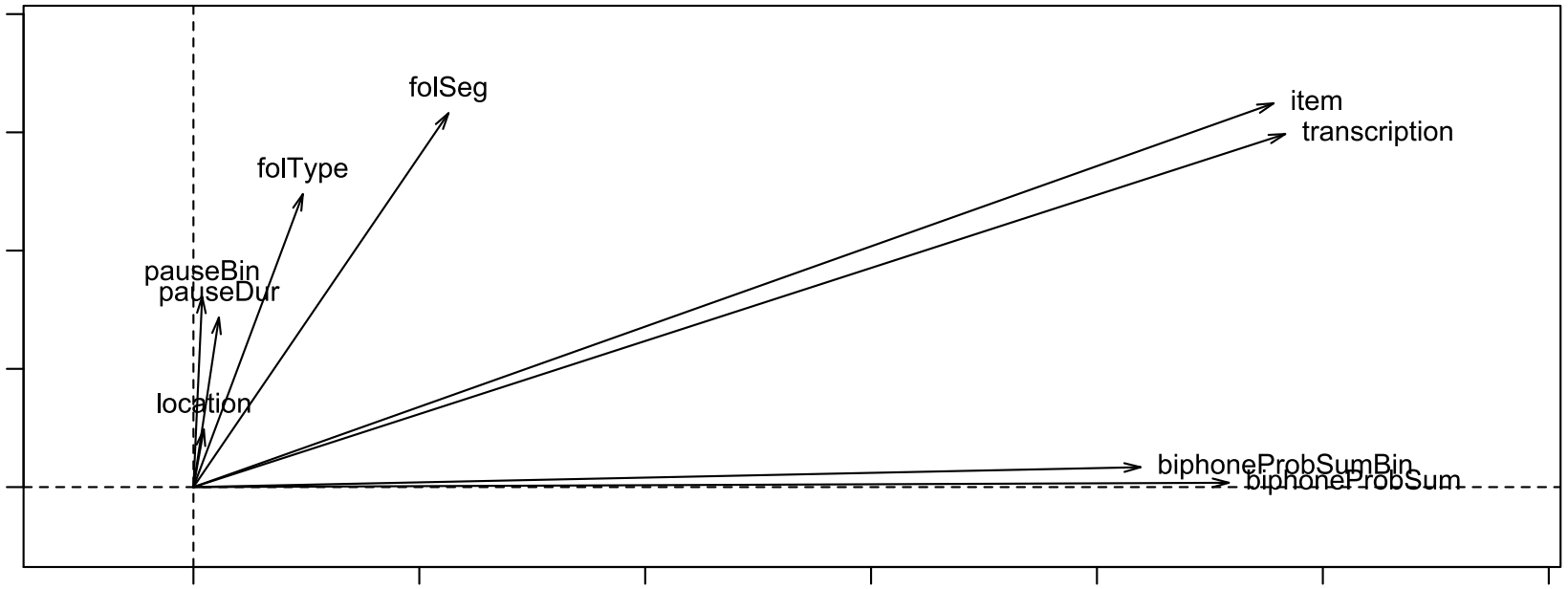
# Kollinearität

- Dabei finden wir folgende starke Korrelationen, d.h. $-0.5 > r > 0.5$

| Variable 1 | Variable 2 | Korrelation |
|---|---|---|
| ITEM | TRANSCRIPTION | 0.82 |
| ITEM | BIPHONEPROBSUM | 0.62 |
| ITEM | BIPHONEPROBSUMBIN | -0.55 |
| SPEAKER | LOCATION | -0.57 |
| FOLSEG | FOLTYPE | -0.86 |
| TRANSCRIPTION | BIPHONEPROBSUM | 0.73 |
| BIPHONEPROBSUM | BIPHONEPROBSUMBIN | -0.69 |
| PAUSEDUR | PAUSEBIN | 0.89 |

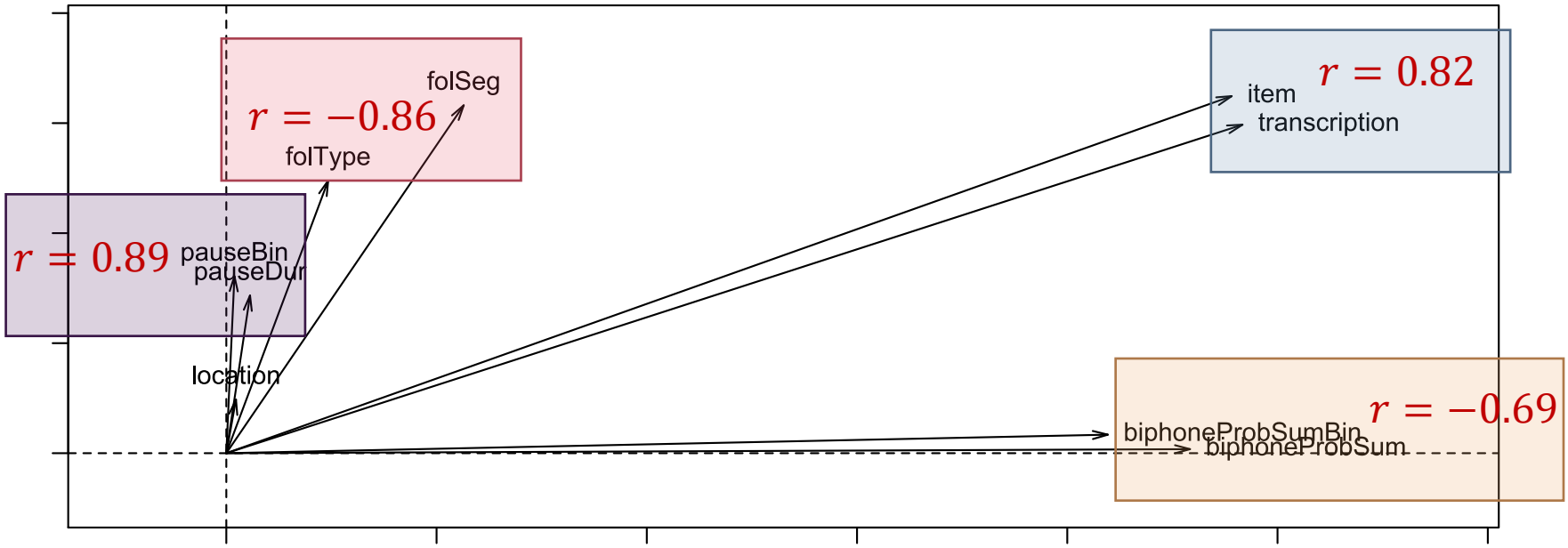
# Kollinearität

- Schauen wir uns die stark korrelierten Variablen und ihre Effekte an:



# Kollinearität

- Schauen wir uns die stark korrelierten Variablen und ihre Effekte an:



# Kollinearität

- Nun haben wir u.a. folgende Möglichkeit
  1. **Vergleich und Ausschluss**: Die stark korrelierten Variablen in simple Modelle einsetzen und diese vergleichen. Die Variable mit der größeren Vorhersagekraft wird behalten.

     **Potentielles Problem**: Der Verlust von Variablen kann einem Verlust an Vorhersagekraft entsprechen.
  2. **Principal Component Analysis**: Ein Verfahren mit welchem aus ursprünglich korrelierten Variablen neue, nicht korrelierte Variablen erstellt werden.

     **Potentielles Problem**: Kompliziert in jeglicher Hinsicht.
- Eine „perfekte“ Lösung gibt es leider nicht – daher schauen wir uns diese Möglichkeit dennoch an.

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:
  1. ITEM VS. TRANSCRIPTION | ITEM VS. BIPHONEPROBSUM | ITEM VS. BIPHONEPROBSUMBIN
  2. TRANSCRIPTION VS. BIPHONEPROBSUM
  3. BIPHONEPROBSUM VS. BIPHONEPROBSUMBIN
  4. SPEAKER VS. LOCATION
  5. FOLSEG VS. FOLTYPE
  6. PAUSEDUR VS. PAUSEBIN

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:
  1. ITEM VS. TRANSCRIPTION | ITEM VS. **BIPHONEPROBSUM** | ITEM VS. **BIPHONEPROBSUMBIN**
  2. TRANSCRIPTION VS. **BIPHONEPROBSUM**
  3. **BIPHONEPROBSUM** VS. **BIPHONEPROBSUMBIN**
  4. SPEAKER VS. LOCATION
  5. FOLSEG VS. FOLTYPE
  6. PAUSEDUR VS. PAUSEBIN

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:
  1. ITEM VS. TRANSCRIPTION | ITEM VS. BIPHONEPROBSUM | ITEM VS. BIPHONEPROBSUMBIN
  2. TRANSCRIPTION VS. BIPHONEPROBSUM
  3. BIPHONEPROBSUM VS. BIPHONEPROBSUMBIN
  4. SPEAKER VS. LOCATION
  5. FOLSEG VS. FOLTYPE
  6. PAUSEDUR VS. PAUSEBIN

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:
  1. ITEM VS. TRANSCRIPTION | ITEM VS. BIPHONEPROBSUM | ITEM VS. BIPHONEPROBSUMBIN
  2. TRANSCRIPTION VS. BIPHONEPROBSUM
  3. BIPHONEPROBSUM VS. BIPHONEPROBSUMBIN
  4. SPEAKER VS. LOCATION
  5. FOLSEG VS. FOLTYPE
  6. PAUSEDUR VS. PAUSEBIN

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:

1. ITEM VS. TRANSCRIPTION | ITEM VS. BIPHONEPROBSUM | ITEM VS. BIPHONEPROBSUMBIN
2. TRANSCRIPTION VS. BIPHONEPROBSUM
3. BIPHONEPROBSUM VS. BIPHONEPROBSUMBIN
4. SPEAKER VS. LOCATION
5. FOLSEG VS. FOLTYPE
6. PAUSEDUR VS. PAUSEBIN

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:
  1. BIPHONEPROBSUM VS. BIPHONEPROBSUMBIN
  - **Potentielle Ergebnisse:**
    - signifikanter Unterschied, d.h. bedeutend verschiedener Fit der beiden Modelle
      = wir behalten die bessere Variable und schließen die andere aus
    - kein signifikanter Unterschied, d.h. beide Variablen modellieren /s/ Dauer gleich gut
      = wir überlegen, welche Variable konzeptionell besser ist/hören auf den AIC-Wert

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:
  1. BIPHONEPROBSUM VS. BIPHONEPROBSUMBIN
  - **Ergebniss:**
    - kein signifikanter Unterschied, d.h. beide Variablen modellieren /s/ Dauer gleich gut = wir entscheiden uns für **biphoneProbSumBin**, da die Interpretation konzeptionell einfacher ist

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:

1. ITEM VS. TRANSCRIPTION | ~~ITEM VS. BIPHONEPROBSUM~~ | ITEM VS. BIPHONEPROBSUMBIN
2. ~~TRANSCRIPTION VS. BIPHONEPROBSUM~~
3. ~~BIPHONEPROBSUM VS. BIPHONEPROBSUMBIN~~
4. SPEAKER VS. LOCATION
5. FOLSEG VS. FOLTYPE
6. PAUSEDUR VS. PAUSEBIN

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:

  2. ITEM VS. TRANSCRIPTION

  - **Ergebniss:**

    - signifikanter Unterschied, d.h. bedeutend verschiedener Fit der beiden Modelle = **item**, modelliert die Dauer von /s/ besser, also behalten wir die Variable

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:
  1. ITEM VS. ~~TRANSCRIPTION~~ | ~~ITEM VS. BIPHONEPROBSUM~~ | ITEM VS. BIPHONEPROBSUMBIN
  2. ~~TRANSCRIPTION VS. BIPHONEPROBSUM~~
  3. ~~BIPHONEPROBSUM VS. BIPHONEPROBSUMBIN~~
  4. SPEAKER VS. LOCATION
  5. FOLSEG VS. FOLTYPE
  6. PAUSEDUR VS. PAUSEBIN

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:

  3. ITEM VS. BIPHONEPROBSUMBIN

  - **Ergebniss:**

    - kein signifikanter Unterschied, d.h. beide Variablen modellieren /s/ Dauer gleich gut = wir entscheiden uns für **biphoneProbSumBin**, da die Interpretation konzeptionell einfacher ist

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:

1. ~~ITEM VS. TRANSCRIPTION~~ | ~~ITEM VS. BIPHONEPROBSUM~~ | ~~ITEM VS. BIPHONEPROBSUMBIN~~
2. ~~TRANSCRIPTION VS. BIPHONEPROBSUM~~
3. ~~BIPHONEPROBSUM VS. BIPHONEPROBSUMBIN~~
4. SPEAKER VS. LOCATION
5. FOLSEG VS. FOLTYPE
6. PAUSEDUR VS. PAUSEBIN

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:

4. SPEAKER VS. LOCATION

- **Ergebniss:**
  - kein signifikanter Unterschied, d.h. beide Variablen modellieren /s/ Dauer gleich gut
    = wir entscheiden uns für **location**, da die Interpretation konzeptionell einfacher ist

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:

1. ~~ITEM VS. TRANSCRIPTION~~ | ~~ITEM VS. BIPHONEPROBSUM~~ | ~~ITEM VS. BIPHONEPROBSUMBIN~~
2. ~~TRANSCRIPTION VS. BIPHONEPROBSUM~~
3. ~~BIPHONEPROBSUM VS. BIPHONEPROBSUMBIN~~
4. ~~SPEAKER VS.~~ LOCATION
5. FOLSEG VS. FOLTYPE
6. PAUSEDUR VS. PAUSEBIN

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:
  4. FOLSEG VS. FOLTYPE
  - **Ergebniss:**
    - kein signifikanter Unterschied, d.h. beide Variablen modellieren /s/ Dauer gleich gut
      = wir entscheiden uns für **folType**, da die Interpretation konzeptionell einfacher ist

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:

1. ~~ITEM VS. TRANSCRIPTION~~ | ~~ITEM VS. BIPHONEPROBSUM~~ | ~~ITEM VS. BIPHONEPROBSUMBIN~~
2. ~~TRANSCRIPTION VS. BIPHONEPROBSUM~~
3. ~~BIPHONEPROBSUM VS. BIPHONEPROBSUMBIN~~
4. ~~SPEAKER VS.~~ LOCATION
5. ~~FOLSEG VS.~~ FOLTYPE
6. PAUSEDUR VS. PAUSEBIN

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:
  4. PAUSEDUR VS. PAUSEBIN
  - **Ergebniss:**
    - kein signifikanter Unterschied, d.h. beide Variablen modellieren /s/ Dauer gleich gut
      = wir entscheiden uns für **pauseBin**, da die Interpretation konzeptionell einfacher ist

# 1. Vergleich und Ausschluss

- anhand der gefundenen Korrelationen müssen wir folgende Vergleiche vornehmen:

1. ~~ITEM VS. TRANSCRIPTION~~ | ~~ITEM VS. BIPHONEPROBSUM~~ | ~~ITEM VS. BIPHONEPROBSUMBIN~~
2. ~~TRANSCRIPTION VS. BIPHONEPROBSUM~~
3. ~~BIPHONEPROBSUM VS. BIPHONEPROBSUMBIN~~
4. ~~SPEAKER VS.~~ LOCATION
5. ~~FOLSEG VS.~~ FOLTYPE
6. ~~PAUSEDUR VS.~~ PAUSEBIN

# 1. Vergleich und Ausschluss

- Es bleiben abschließend folgende Variablen zur Erstellung eines vollen Modells übrig:
  1. BIPHONEPROBSUMBIN
  2. LOCATION
  3. FOLTYPE
  4. PAUSEBIN

- Zusätzlich bleiben natürlich noch die Variablen im Spiel, welche keine starken Korrelationen vorgewiesen haben:
  - GENDER, TYPEOFS, SLIDENUMBER, BASEDUR, AGE, MONOMULTILINGUAL, SPEAKINGRATE, PREC

# 1. Vergleich und Ausschluss

- Das volle Modell enthält also folgende Variablen:

  BIPHONEPROBSUMBIN, LOCATION, FOLTYPE, PAUSEBIN, GENDER, TYPEOFS, SLIDENUMBER, BASEDURLOG, AGE, MONOMULTILINGUAL, SPEAKINGRATE, PREC

- Das finale Modell (nach `step()`) enthält folgende Variablen

  BIPHONEPROBSUMBIN, LOCATION, PAUSEBIN, GENDER, TYPEOFS, BASEDURLOG, MONOMULTILINGUAL, SPEAKINGRATE

- Keine dieser Variablen sind kollinear, d.h. die Berechnung der Koeffizienten, sowie die Signifikanzlevel der einzelnen Variablen sind als verlässlich einzustufen

- Allerdings haben wir auf dem Weg zum finalen Modell knapp die Hälfte unserer Variablen verloren...

# 2. Principal Component Analysis

- Principal Component Analysis – PCA – beschreibt ein Verfahren mit welchem aus ursprünglich korrelierten Variablen neue, nicht korrelierte Variablen erstellt werden
- Wir schauen uns dieses Verfahren anhand fiktiver Daten an
- Hierzu erstellen wir zunächst ein Data Frame mit den problematischen Variablen:

```
var1, var2, var3, var4, var5, var6, var7, var8, var9
```

- Jede dieser Variablen ist mit mindestens einer der anderen Variablen stark korreliert

# 2. Principal Component Analysis

- Nun nutzen wir die PCA-Funktion, die standardmäßig in R vorhanden ist:

```
pc <- princomp(df, cor=TRUE, score=TRUE)
```

# 2. Principal Component Analysis

- Danach können wir uns die s.g. **Eigenwerte / Eigenvalues** der neu erstellen Variablen anschauen:

```
> library("factoextra")
> get_eigenvalue(pc)
       eigenvalue variance.percent  cumulative.variance.percent
Dim.1 2.71706780        30.1896422                     30.18964
...
```

# 2. Principal Component Analysis

- Je höher der Eigenwert einer erstellten Variable ist, desto mehr Varianz kann sie in den ursprünglichen Daten erklären.

| | eigenvalue | variance.percent | cumulative.variance.percent |
|---|---|---|---|
| Dim.1 | 2.71706780 | 30.1896422 | 30.18964 |
| Dim.2 | 2.62128516 | 29.1253906 | 59.31503 |
| Dim.3 | 2.42922025 | 26.9913361 | 86.30637 |
| Dim.4 | 0.36751103 | 4.0834559 | 90.38982 |
| Dim.5 | 0.33664969 | 3.7405521 | 94.13038 |
| Dim.6 | 0.27992699 | 3.1102999 | 97.24068 |
| Dim.7 | 0.08998043 | 0.9997825 | 98.24046 |
| Dim.8 | 0.08094112 | 0.8993458 | 99.13981 |
| Dim.9 | 0.07741752 | 0.8601947 | 100.00000 |

# 2. Principal Component Analysis

- Schließlich müssen wir entscheiden, welche neu erstellen Variablen wir behalten und nutzen möchten. Hierzu gibt es einige Richtlinien, an welchen man sich orientieren kann:
    1. Der Eigenwert einer Variable muss mindestens 1 betragen, da eine Variable mit Eigenwert 1 mindestens sich selbst erklären kann.
    2. Die ‚cumulative variance explained' der neuen Variablen sollte mindestens 70 % betragen
    3. Es sollte keine großen Unterschiede zwischen den Eigenwerten der neuen Variablen geben
    4. Neue Variablen sind nur hilfreich, wenn sie auch interpretierbar sind

# 2. Principal Component Analysis

- Schließlich müssen wir entscheiden, welche neu erstellen Variablen wir behalten möchten.

Eigenwert >= 1

cumu. variance explained >= 70%

| | eigenvalue | variance.percent | cumulative.variance.percent |
|---|---|---|---|
| Dim.1 | 2.71706780 | 30.1896422 | 30.18964 |
| Dim.2 | 2.62128516 | 29.1253906 | 59.31503 |
| Dim.3 | 2.42922025 | 26.9913361 | 86.30637 |
| Dim.4 | 0.36751103 | 4.0834559 | 90.38982 |
| Dim.5 | 0.33306469 | 3.7405521 | 94.13038 |
| Dim.6 | 0.27992699 | 3.1102999 | 97.24068 |
| Dim.7 | 0.08998043 | 0.9997825 | 98.24046 |
| Dim.8 | 0.08094112 | 0.8993458 | 99.13981 |
| Dim.9 | 0.07741752 | 0.8601947 | 100.00000 |

keine großen Sprünge

# 2. Principal Component Analysis

- Wir behalten also PCA Components 1, 2 und 3 als neue Variablen:

```
pcframe <- as.data.frame(pc$scores)[1:3]
```

# 2. Principal Component Analysis

- Doch was bedeuten die neuen Variablen? Das stellen wir anhand ihrer s.g. Loadings fest:

```
> pc$loadings[,1:3]
```

```
Loadings:
      Comp.1 Comp.2 Comp.3
var1   0.316  0.168  0.414
var2  -0.376 -0.185 -0.435
var3  -0.353 -0.191 -0.417
var4         -0.474  0.260
var5          0.517 -0.301
var6          0.498 -0.285
var7  -0.433  0.222  0.273
var8   0.465 -0.251 -0.279
var9   0.458 -0.221 -0.272
```

Die Loadings einer PCA Component stellen ihre **Korrelation** mit den ursprünglichen Variablen dar. Je höher die Korrelation, desto mehr der ursprünglichen Variable ist in einer Komponente enthalten.

# 2. Principal Component Analysis

- Doch was bedeuten die neuen Variablen? Das stellen wir anhand ihrer s.g. Loadings fest:

```
> pc$loadings[,1:3]

Loadings:
      Comp.1 Comp.2 Comp.3
var1   0.316  0.168  0.414
var2  -0.376 -0.185 -0.435
var3  -0.353 -0.191 -0.417
var4         -0.474  0.260
var5          0.517 -0.301
var6          0.498 -0.285
var7  -0.433  0.222  0.273
var8   0.465 -0.251 -0.279
var9   0.458 -0.221 -0.272
```

Unsere drei Komponenten spiegeln die ursprünglichen Variablen gut wider:

Comp.1 umfasst die Effekte von var7, var8 und var9

Comp.2 umfasst die Effekte von var4, var5 und var6

Comp.3 umfasst die Effekte von var1, var2 und var3

# 2. Principal Component Analysis

- Doch was bedeuten die neuen Variablen? Das stellen wir anhand ihrer s.g. Loadings fest:

```
> pc$loadings[,1:3]

Loadings:
      Comp.1 Comp.2 Comp.3
var1   0.316  0.168  0.414
var2  -0.376 -0.185 -0.435
var3  -0.353 -0.191 -0.417
var4         -0.474  0.260
var5          0.517 -0.301
var6          0.498 -0.285
var7  -0.433  0.222  0.273
var8   0.465 -0.251 -0.279
var9   0.458 -0.221 -0.272
```

Linguistisches Beispiel:

Nehmen wir an, dass
- var7 = speech rate
- var8 = phon. neighbourhood
- var9 = orth. neighbourhood

Comp.1 zeigt uns, dass der Effekt von var7 in die entgegengesetzte Richtung zu den Effekten von var8 und var9 geht

# 2. Principal Component Analysis

- Doch was bedeuten die neuen Variablen? Das stellen wir anhand ihrer s.g. Loadings fest:

```
> pc$loadings[,1:3]

Loadings:
      Comp.1 Comp.2 Comp.3
var1   0.316  0.168  0.414
var2  -0.376 -0.185 -0.435
var3  -0.353 -0.191 -0.417
var4         -0.474  0.260
var5          0.517 -0.301
var6          0.498 -0.285
var7  -0.433  0.222  0.273
var8   0.465 -0.251 -0.279
var9   0.458 -0.221 -0.272
```

Linguistisches Beispiel:

Je nach Vorzeichen des Koeffizienten von Comp.1 in einer Regressionsanalyse finden wir daher:

1. positiv: höhere speech rate = niedrigerer Wert der abhängigen Variable
2. negativ: höhere speech rate = höherer Wert der abhängigen Variable

# Ja, PCAs sind kompliziert...

# Kollinearität und andere Probleme

- Bisher haben wir relativ naiv Modelle erstellt
  - irgendeine Variable wird vorhergesagt
  - irgendwelche Variablen sagen vorher
- Problem: Potentielle Gefahren
  1. Nicht-Normalverteilung der Variablen ✓
  2. Kollinearität ✓
  3. Interaktionen
  4. Geringe Datenmenge / zu weniger „Power“