Inhaltsverzeichnis

1 Multiple Regression in R

StatsReminder

Im Regressionskontext haben wir bisher nur den Zusammenhang zwischen zwei Variablen betrachtet. Psychologische Theorien sind meist komplexer. Hypothesen, die wir überprüfen wollen, beinhalten meist den Zusammenhang zwischen mehr als zwei Variablen (sowie teilweise Interaktionen zwischen diesen Variablen, s. moderierte Regression unten). Die multiple lineare Regression ist ein statistisches Werkzeug für die Modellierung der Beziehung einer abhängigen Variablen mit mehreren unabhängigen Variablen. Im additiven Fall (keiner Interaktion):

$$\widehat{Y} = b_0 + b_1 X_1 + \ldots + b_n X_n$$

1.1 Vorbereitung

```
load("dat/erstis_neu.RData")
```

Wir erstellen uns erneut der Einfachheit halber eine reduzierte Version des Datensatzes. Wir werden diesmal eine multiple Regression zur Vorhersage von Lebenszufriedenheit (`lz.1`) durch die Stimmungsdimensionen "gute vs. schlechte Stimmung" (valence; `gs.1`) und "wach vs. müde" (energetic arousal; `wm.1`) betrachten.

```
sub <- na.omit(erstis[, c("lz.1","gs.1","wm.1")])
```

Schauen wir uns zunächst die Korrelationsmatrix des reduzierten Datensatzes an:

```
round(cor(sub), digits = 3)
```

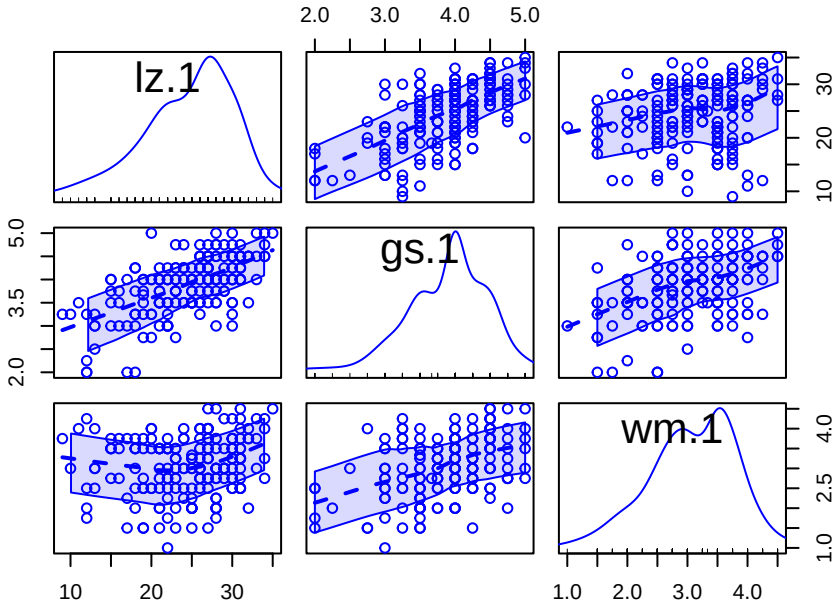
```
      lz.1  gs.1  wm.1
lz.1 1.000 0.626 0.159
gs.1 0.626 1.000 0.427
wm.1 0.159 0.427 1.000
```

- Valence korreliert stark positiv mit Lebenszufriedenheit
- Energetic arousal korreliert schwach positiv mit Lebenszufriedenheit
- Die beiden Stimmungsdimensionsn (*valence* und *energetic arousal*) korrelieren mittel bis stark positiv

Bevor wir eine lineare Regression berechnen, ist es sinnvoll, sich die (bivariaten) Verteilungen der Variablen anzuschauen. Dies kann einen Eindruck darüber liefern, ob die Variablen ansatzweise normalverteilt sind und ob der bivariate Zusammenhang der Variablen annähernd linear (im Gegensatz zu z.B. quadratisch) ist. Dies ist relevant für die Annahmen welche der (Inferenzstatistik der) linearen Regression zugrunde liegen (s. kommendes Sommersemester).

Mit der Funktion `scatterplotMatrix` aus dem `car`-Paket kann man auf sehr zugängliche Art und Weise univariate und bivariate Zusammenhänge grafisch darstellen:

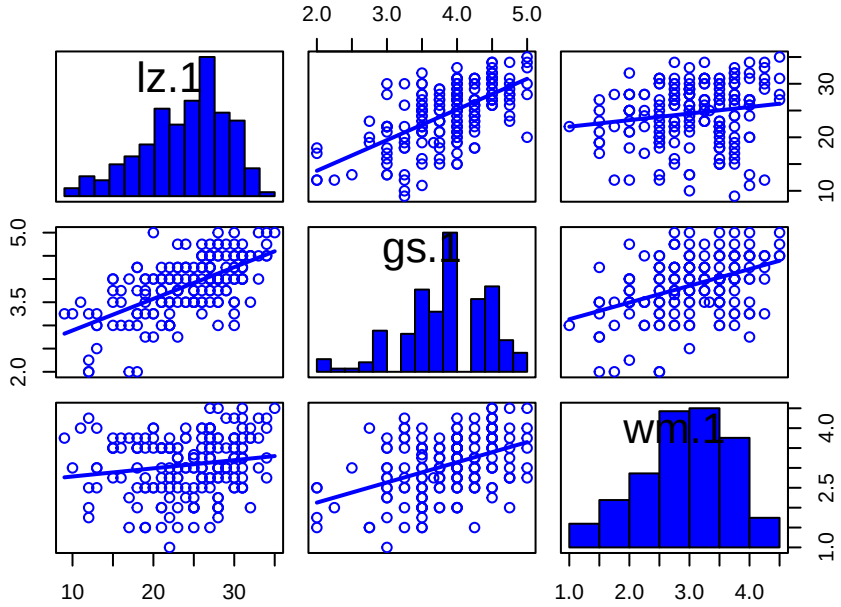
```
library(car)
scatterplotMatrix(sub, regLine = F)
```



- Auf der Diagonalen sind die Schätzungen für die univariaten Verteilungen der einzelnen Variablen zu finden (siehe Kerndichteschätzer)
- Abseits der Diagonalen sind die bivariaten Streudiagramme inkl. einer sogenannten *loess line* der Variablen zu finden (mittlere gestrichelte Linie; siehe Loess Anpassungslinie)
- Die bivariaten Zusammenhänge von *valence* und *energetic arousal* mit Lebenszufriedenheit sind annähernd linear (diese Information kann man der ersten Zeile (oder auch der ersten Spalte) entnehmen). Der quadratisch aussehende Zusammenhang von `lz.1` und `gs.1` links unten (3. Zeile, 1. Spalte) ist vermutlich auf die beiden Werte mit sehr geringer sowie sehr hoher Wachheit zurückzuführen.

Alternativ können die Verteilungen bivariater Zusammenhänge mit der `scatterplotMatrix` Funktion auch durch Histogramme und Streudiagramme mit einer linearen Regressionslinie dargestellt werden:

```
scatterplotMatrix(sub, smooth = F, diagonal = list(method ="histogram"))
```



1.1.1 Exkurs Kerndichteschätzer und Loess Anpassungslinie (nicht klausurrelevant)

1.1.1.1 Kerndichteschätzer Der Kerndichteschätzer ist ein Verfahren zur Bestimmung der Verteilung einer oder mehrerer Variablen. Im Gegensatz zum Histogramm erhält man beim Kerndichteschätzer eine stetige Verteilung. Durch die Wahl eines geeigeneten Kerns, welcher eine Gewichtung der beobachteten Werte um einen Wert der x-Achse zur Schätzung der Verteilung vornimmt, wird die Häufigkeitsverteilung (s. Histogramm) sozusagen geglättet.

1.1.1.2 Loess Anpassungslinie Loess steht für *locally estimated scatterplot smoothing* (manchmal auch lowess: *locally weighted scatterplot smoother*) und fällt unter die sogenannten Glättungsverfahren. Im Gegensatz zu einer linearen Regression kann man mit Loess eine kurvige Linie für den Zusammenhang zweier Variablen schätzen. Es wird lokal (das heißt, in einem Subset der Daten, die in einem Fenster mit einer vorher spezifizierten Breite um einen x-Wert der Variable auf der x-Achse liegen) eine Regression geschätzt. Dies wird sukzessive für verschiedene (teilweise alle) Werte auf der x-Achse durchgeführt und die lokal vorhergesagten Werte werden genutzt, um eine Linie anzupassen. Anhand der Loess-Linie kann die Form des Zusammenhangs der Variablen eingeschätzt werden (z.B. ob linear oder annähernd linear?).

1.1.2 Zentrierung der Prädiktoren

Keiner der Prädiktoren umfasst in seinem Wertebereich die Null. Daher sollten die Variablen vorab zentriert werden, damit eine inhaltlich sinnvolle Interpretation des *Intercepts* ermöglicht wird.

```
sub$wm.1_c <- sub$wm.1 - mean(sub$wm.1)
sub$gs.1_c <- sub$gs.1 - mean(sub$gs.1)
```

Da der von der `scale()` Funktion erstellte Datentyp der Variablen nicht mit der später verwendeten plot-Funktion kompatibel ist, nutzen wir hier die "klassische" Variante der Zentrierung. Wir können uns noch einmal versichern, ob die Zentrierung erfolgreich war:

```
mean(sub$wm.1_c)
```

```
[1] 2.820302e-17
```

```
mean(sub$gs.1_c)
```

```
[1] -2.366027e-18
```

```
sd(sub$wm.1_c)
```

```
[1] 0.7275496
```

```
sd(sub$gs.1_c)
```

```
[1] 0.6133199
```

- Die erstellten Variablen haben einen Mittelwert von 0 und ihre ursprüngliche Standardabweichung.

1.2 `lm()`

Um ein multiples Regressionsmodell zu schätzen, benutzen wir erneut die Funktion `lm()`. Der einzige Unterschied in der Durchführung zu einer einfachen linearen Regression besteht bei der multiplen Regression in der übergebenen Formel.

```
mr <- lm(lz.1 ~ gs.1_c + wm.1_c, data = sub)
```

- Die unabhängigen Variablen werden mit `+` innerhalb von `lm()` verknüpft
  - Allgemeine Formel: `av ~ uv_1 + ... + uv_i`
  - `av` = abhängige Variable
  - `uv` = unabhängige Variable

Die Regressionskoeffizienten können wieder mit `coef()` abgerufen werden:

```
coef(mr)
```

```
(Intercept)      gs.1_c      wm.1_c
  24.534392    6.250610   -1.026558
```

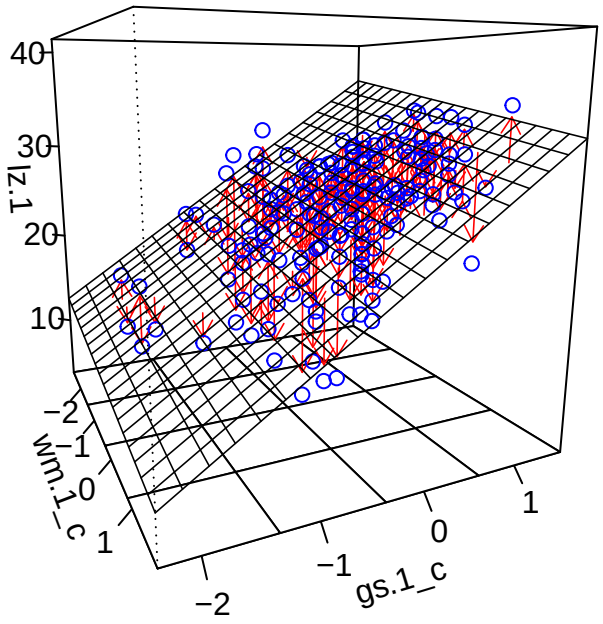
- $b_0 = 24.54$ (`Intercept`): Erwarteter Lebenszufriedenheitswert für Personen mit durchschnittlicher Ausprägung auf beiden Stimmungsdimensionen (wir haben die unabhängigen Variablen zentriert!)
- *Valence* (gute Stimmung) hat einen positiven Effekt auf die Lebenszufriedenheit, *energetic arousal* (Wachheit) hat einen negativen Effekt auf die Lebenszufriedenheit
- $b_{\text{gs}} = 6.25$ (`Slope` für *valence*): Unterschied in der geschätzten Lebenszufriedenheit zwischen zwei Personen, die sich um eine Einheit in ihrer guten / schlechten Stimmung (*valence*) unterscheiden, sich aber nicht in ihrer Wachheit unterscheiden
- $b_{\text{wm}} = -1.03$ (`Slope` für *energetic arousal*): Unterschied in der geschätzten Lebenszufriedenheit zwischen zwei Personen, die sich um eine Einheit in ihrer Wachheit voneinander unterscheiden, aber nicht in der guten / schlechten Stimmung

1.3 Darstellung der Regressionsebene

Es besteht auch die Möglichkeit, eine multiple Regression mit zwei unabhängigen Variablen dreidimensional zu visualisieren:

```
library(rockchalk)

plotPlane(mr, plotx2 = "gs.1_c", plotx1 = "wm.1_c",    # notwendige Argumente
          drawArrows = T, phi = 12, theta = 65,        # optionale Grafikparameter
          ticktype = "detailed", llty = 1, plwd = 1, alty = 1, alwd = 0.8, llwd = 0.8)
```



- Die Argumente `phi` und `theta` bestimmen aus welchem Winkel die Grafik geplottet wird
- `drawArrows` gibt an, ob die roten Pfeile, welche die Entfernung und Richtung der Punkte von der Ebene markieren, eingezeichnet werden sollen oder nicht

Es gibt auch die Möglichkeit, eine drehbare dreidimensionale Regressionsebene in R zu erzeugen. Führen Sie dazu die folgenden Befehle aus. Es öffnet sich in R ein neues Fenster mit dem Plot.

```
if (!require(rgl)) install.packages("rgl")
if (!require(car)) install.packages("car")
library(car)
scatter3d(lz.1 ~ wm.1_c+gs.1_c, data= sub)
```

1.4 Multipler Determinationskoeffzient

StatsReminder

Der multiple Determinationskoeffizient ist definiert als der Anteil der durch die Regression erklärten Varianz $s^2_{\hat{Y}}$ an der Gesamtvarianz der abhängigen Variablen s^2_Y (Anteil der durch alle Prädiktoren gemeinsam erklärten Unterschiedlichkeit in der abhängigen Variablen):

$$R^2 = \frac{\sum(\hat{y}_i - \bar{y})}{\sum(y_i - \bar{y})} = \frac{s^2_{\hat{Y}}}{s^2_Y}$$

Der Determinationskoeffizient kann mit der `summary`-Funktion abgefragt werden:

```
summary(mr)$r.squared
```

```
[1] 0.4057984
```

- Rund 40,6 % der Unterschiede in der Lebenszufriedenheit können auf Unterschiede in den beiden Stimmungsdimensionen zurückgeführt werden (durch das Modell erklärt werden).

Die multiple Korrelation R ist die Wurzel aus diesem Wert:

```
sqrt(summary(mr)$r.squared)
```

```
[1] 0.6370231
```

1.5 Nützlichkeit

Die Nützlichkeit ist definiert als Differenz zwischen den Determinationskoeffizienten zweier Regressionsmodelle, wobei sich die Modelle nur um die Hinzunahme eines einzigen Prädiktors unterscheiden dürfen. Zur Berechnung der Nützlichkeit müssen zunächst Vergleichsmodelle geschätzt werden. Im Folgenden schauen wir uns das Inkrement in R^2 zwischen dem bereits bekannten Regressionsmodell `mr` und demselben Regressionsmodell ohne den Prädiktor `energetic arousal` an (Nützlichkeit von `energetic arousal`):

```
m_gs <- lm(lz.1 ~ gs.1_c, data = sub)
summary(mr)$r.squared - summary(m_gs)$r.squared
```

```
[1] 0.0144453
```

- Varianzaufklärung des Modells mit *energetic arousal* - Varianzaufklärung des Modells ohne *energetic arousal*
- Die Wachheit (energetic arousal) kann über *valence* hinaus rund 1,4 % der Unterschiede in der Lebenszufriedenheit erklären

Für *valence* kann die Nützlichkeit auf die gleiche Weise bestimmt werden.

2 Moderierte Regression in R

StatsReminder

Eine moderierte Regression ziehen wir heran, wenn wir davon ausgehen, dass der Effekt einer unabhängigen Variable auf die abhängige Variable von der Ausprägung einer anderen unabhängigen Variable abhängt. Praktisch heißt das, dass wir eine Interaktion (Produkt der beiden unabhängigen Variablen) in das multiple Regressionsmodell mit aufnehmen:

$$\widehat{Y} = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_1 X_2$$

Das Wort *moderiert* soll darauf hinweisen, dass die Ausprägung eines Prädiktors (des sogenannten Moderators) den Effekt des anderen Prädiktors auf die abhängige Variable verändert. **Achtung!** Aus rein statistischer Sicht ist die Interaktion zwischen zwei Variablen symmetrisch. Welche Variable wir als *Moderator* bezeichnen ist damit eine theoretisch inhaltliche Frage.

2.1 `lm()`

Nun möchten wir untersuchen, ob sich der Effekt der guten Stimmung auf die Lebenszufriedenheit in Abhängigkeit der Wachheit verändert, ob also die gute Stimmung und die Wachheit in Wechselwirkung stehen in ihrem Effekt auf die Lebenszufriedenheit. Regressionsgleichung:

$$lz = b_0 + b_1 \cdot gs + b_2 \cdot wm + \underbrace{b_3 \cdot gs \cdot wm}_{\text{Interaktionsterm}} + E$$

Um eine moderierte Regression zu schätzen, muss der `lm()`-Funktion ein Interaktionsterm übergeben werden. Dafür werden die relevanten Prädiktoren im `lm()` Befehl mit `*` verknüpft. Dabei wird der Produktterm zusätzlich zu den zentrierten Prädiktoren in die Regressionsgleichung aufgenommen, ohne dass explizit eine Produktvariable im Datensatz erstellt werden müsste.

```
m_int <- lm(lz.1 ~ wm.1_c*gs.1_c, sub)
```

Auf das resultierende Modellobjekt `m_int` können nun alle bereits kennengelernten Funktionen angewendet werden. So können wir die Regressionskoeffizienten mit `coef()` abrufen:

```
round(coef(m_int), 2)
```

```
  (Intercept)         wm.1_c         gs.1_c wm.1_c:gs.1_c
        24.30          -0.91           6.47           1.26
```

```
summary(m_int)$r.squared
```

```
[1] 0.4182535
```

- Regressionsgleichung:

$$\widehat{lz} = 24.30 + 6.47 \cdot gs - 0.91 \cdot wm + 1.26 \cdot gs \cdot wm$$

Das Modell kann rund 42 Prozent der Unterschiede in der Lebenszufriedenheit vorhersagen ($R^2 = 0.418$).

- $b_0 = 24.30$ (`Intercept`): Geschätzter Lebenszufriedenheitswert für Personen mit durchschnittlicher Ausprägung auf beiden unabhängigen Variablen
- $b_{\text{g-Stim}} = 6.47$ (`Slope` für gute Stimmung): Unterschied in der geschätzten Lebenszufriedenheit zwischen zwei Personen mit mittlerer wacher-müder Stimmung, die sich um eine Einheit in guter Stimmung unterscheiden
- $b_{\text{wm-Stim}} = -0.91$ (`Slope` für wache-müde Stimmung): Unterschied in der geschätzten Lebenszufriedenheit zwischen zwei Personen mit mittlerer guter Stimmung, die sich um eine Einheit in der wachen-müden Stimmung unterscheiden

- $b_{\text{g-Stim * wm-Stim}} = 1.26$ (Interaktionsterm): Für höhere Ausprägungen von `gs` ist der Effekt von `wm` auf `lz` weniger stark negativ (bzw. ab sehr hohen Werten dann leicht positiv).

```
summary(m_int)$r.sq - summary(mr)$r.sq
```

```
[1] 0.01245504
```

Durch die Hinzunahme des Interaktionseffekts kann rund 1,2 % mehr Varianz in der Lebenszufriedenheit aufgeklärt werden.

2.1.1 Alternative Schreibweise

Bei mehreren Prädiktoren kann die Schreibweise mit `*` zu ungewollten zusätzlichen Termen (z. B. Dreifachinteraktion) führen. Gezielter könnte man Zweifach-Interaktionen einfügen mit der Schreibweise, wie sie in der `summary()` genutzt wird:

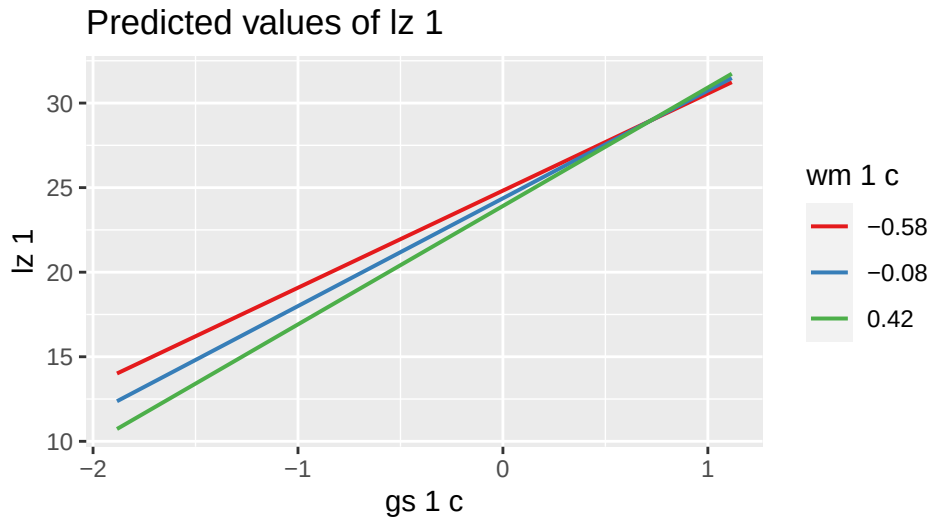
```
m_int <- lm(lz.1 ~ wm.1_c + gs.1_c + wm.1_c:gs.1_c, data = sub)
```

2.2 Darstellung

2.2.1 Bedingte Regressionsgeraden, sog. Simple Slopes

Die bedingten Regressionsgeraden können wir über die `plot_model()` Funktion aus dem Paket `sjPlot` grafisch darstellen. Als Werte des Moderators können wir z.B. die Quartile (`quart2`) wählen. Die Standardeinstellung ist `meansd`, d. h. Mittelwert und Mittelwert plus/minus eine Standardabweichung ($MW \pm 1SD$). Als Moderator wird per Standard der zweite Prädiktor im Argument `terms` gewählt (hier: `wm.1`).

```
library(sjPlot)
plot_model(m_int, type= "pred",
           terms = c("gs.1_c", "wm.1_c[quart2]"),
           ci.lvl = NA)               # graph. Parameter
```



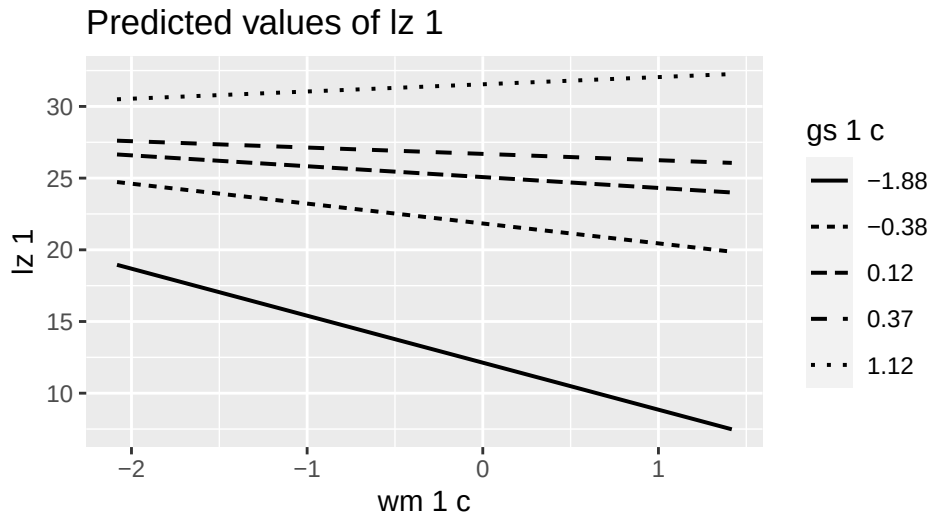
- *Energetic arousal* moderiert den Einfluss der guten Stimmung auf die Lebenszufriedenheit. Der Effekt der guten Stimmung auf die Lebenszufriedenheit ist bei hohem energetic arousal geringfügig stärker.

- Eingesetzt in die Regressionsgleichung:

$$\begin{aligned}
\widehat{lz.1}_{wm.1==Q_3=0.42} &= 24.3 + 6.47 \cdot gs.1 - 0.91 \cdot 0.42 + 1.26 \cdot gs.1 \cdot 0.42 \\
&= (24.3 - 0.38) + (6.47 + 0.53) \cdot gs.1 \\
&= 23.9 + 7 \cdot gs.1 \\
\widehat{lz.1}_{wm.1==Q_2=-0.08} &= 24.3 + 6.47 \cdot gs.1 - 0.91 \cdot (-0.08) + 1.26 \cdot gs.1 \cdot (-0.08) \\
&= (24.3 + 0.07) + (6.47 - 0.10) \cdot gs.1 \\
&= 24.4 + 6.37 \cdot gs.1 \\
\widehat{lz.1}_{wm.1==Q_1=-0.58} &= 24.3 + 6.47 \cdot gs.1 - 0.91 \cdot (-0.58) + 1.26 \cdot gs.1 \cdot (-0.58) \\
&= (24.3 + 0.53) + (6.47 - 0.73) \cdot gs.1 \\
&= 24.8 + 5.74 \cdot gs.1
\end{aligned}$$

Um in der Abbildung zu ändern, welcher Prädiktor als Moderator interpretiert wird, können wir wie folgt die `plot_model()` Argumente ändern:

```
plot_model(m_int, type = "pred",
           terms = c("wm.1_c", "gs.1_c[quart]"), # getauschte Reihenfolge
           ci.lvl = NA, colors = "bw")           # graph. Parameter
```



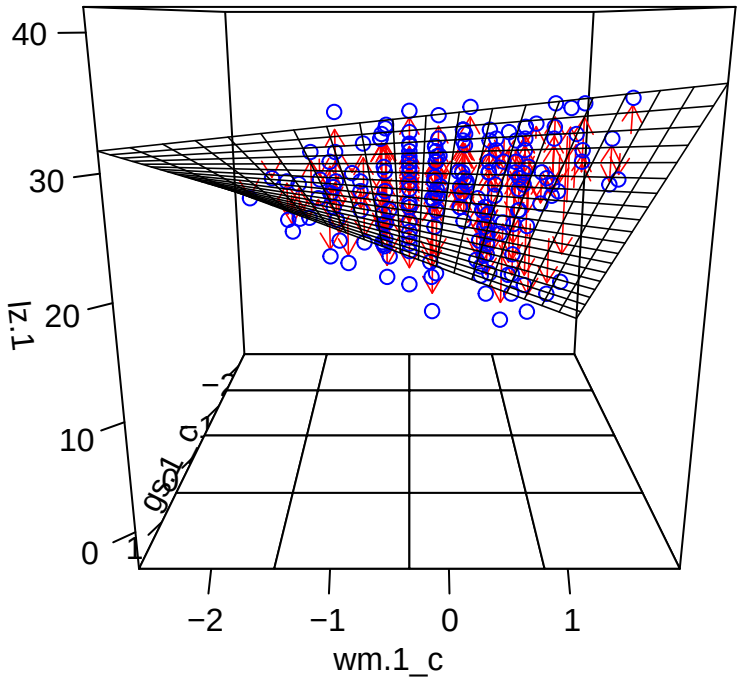
Hinter der Moderatorvariablen (hier: `gs.1_c`) wird die Art der Wertefixierung gewählt. Hier sind es Quantile, d. h. Quartile plus Minimum und Maximum wie in `quantile(flat$gs.1_c)`.

2.3 Darstellung der Regressionsebene

```
library(rockchalk)

m_int <- lm(lz.1 ~ wm.1_c*gs.1_c, sub)

rockchalk::plotPlane(m_int, plotx2 = "wm.1_c", plotx1 = "gs.1_c", # pch = "+",
                     drawArrows = T, phi = 15, theta = 90,  llty = 1,
                     plwd=1, alty=1,alwd=0.8, llwd=0.8,
                     ticktype= "detailed")
```



2.4 Standardisierung

Um die Größe der Effekte besser einschätzen zu können, können standardisierte Regressionsgewichte berechnet werden. Dazu werden *alle* beteiligten numerischen Variablen (inkl. Kriterium) standardisiert und das Modell erneut gerechnet.

```
library(jtools)
sub_z <- standardize(data = sub, vars = c("lz.1", "gs.1", "wm.1"))
m_int_z <- lm(lz.1 ~ gs.1*wm.1, data = sub_z)
coef(m_int_z)
```

```
(Intercept)        gs.1        wm.1   gs.1:wm.1
     -0.043       0.707      -0.118       0.100
```

Der Intercept im Modell mit **standardisierten** Variablen beträgt 0 (Mittelwert der z-standardisierten abhängigen Variable). Die Regressionsgewichte haben sich geändert und beziehen sich nun auf eine Standardabweichung als Einheit. Da sie sich jetzt auf dieselbe Einheit beziehen, kann man sie besser vergleichen und sieht, dass der Effekt von `gs.1` auf `lz.1` stärker ist als der von `wm.1` auf `lz.1`.

3 Übersicht

3.1 Neue wichtige Konzepte

- **Multiple Regression**
- **Determinationskoeffzient**
- **Nützlichkeit**
- **Moderierte Regression**
- **Bedingte Regressionsgeraden**

3.2 Neue wichtige Befehle, Argumente, Operatoren

Funktion	Verwendung
`lm(y ~ x * z)`	Interaktionsterm der Prädiktoren ins Modell einfügen.
`scatterplotMatrix()`	Darstellung bivariater Verteilungen in einer Grafik

3.3 Neue optionale Befehle, Argumente, Operatoren

Funktion	Verwendung
`plot_model()` mit `type = "pred"`	Bedingte Regressionsgeraden darstellen (Paket `sjPlot`)
`plotPlane`	Dreidimensionaler Scatterplot mit Regressionsebene (Paket `rockchalk`)
`scatter3d`	Interaktiver dreidimensionaler Scatterplot mit Regressionsebene (Paket `car`)