

Übungszettel Messwiederholung und Zeitreihen

M.Psy.205, Dozent: Peter Zezula

Joshua Driesen (joshua.driesen@stud.uni-goettingen.de)

Deutsch

Links

[Übungszettel als PDF-Datei zum Drucken](#)

Übungszettel mit Lösungen

[Lösungszettel als PDF-Datei zum Drucken](#)

[Der gesamte Übungszettel als .Rmd-Datei](#) (Zum Downloaden: Rechtsklick > Speichern unter...)

Hinweise zur Bearbeitung

1. Bitte beantworten Sie die Fragen in einer .Rmd Datei. Sie können Sie über `Datei > Neue Datei > R Markdown...` eine neue R Markdown Datei erstellen. Den Text unter dem *Setup Chunk* (ab Zeile 11) können Sie löschen. [Unter diesem Link](#) können Sie auch unsere Vorlage-Datei herunterladen (Rechtsklick > Speichern unter...).
2. Informationen, die Sie für die Bearbeitung benötigen, finden Sie auf der [Website der Veranstaltung](#)
3. Zögern Sie nicht, im Internet nach Lösungen zu suchen. Das effektive Suchen nach Lösungen für R-Probleme im Internet ist tatsächlich eine sehr nützliche Fähigkeit, auch Profis arbeiten auf diese Weise. Die beste Anlaufstelle dafür ist der [R-Bereich der Programmiererplattform Stackoverflow](#)
4. Auf der Website von R Studio finden Sie sehr [hilfreiche Übersichtszettel](#) zu vielen verschiedenen R-bezogenen Themen. Ein guter Anfang ist der [Base R Cheat Sheet](#)

Ressourcen

Da es sich um eine praktische Übung handelt, können wir Ihnen nicht alle neuen Befehle einzeln vorstellen. Stattdessen finden Sie hier Verweise auf sinnvolle Ressourcen, in denen Sie für die Bearbeitung unserer Aufgaben nachschlagen können.

Ressource	Beschreibung
Field, Kap. 13	Für reine Messwiederholungs-ANOVA
Field, Kap. 14	Für Mixed Designs mit sowohl within-, als auch between subjects Faktoren
Field, Kap. 19	Für Multi-Level-Analysen im Allgemeinen

Tipp der Woche

Effizientere Modellformeln

Wenn Sie beispielsweise im `lm()`-Befehl ein Modell mit mehreren Prädiktoren und deren Interaktion aufstellen wollen, können Sie statt `kriterium ~ praediktor1 + praediktor2 + praediktor1:praediktor2` auch einfach `kriterium ~ praediktor1*praediktor2` schreiben, und R übernimmt das Aufdröseln für Sie.

Allgemeines

Diese Woche behandeln wir Situationen, in denen von einer Person (oder einem Land o.ä.) mehrere Messungen in unterschiedlichen Bedingungen/zu unterschiedlichen Zeitpunkten vorliegen. Hierdurch ist in aller Regel eine Annahme der einfachen Regression verletzt, da die Werte einer Person wahrscheinlich recht stark miteinander korrelieren - es besteht also Autokorrelation. Die klassische Lösung ist eine Messwiederholungs-ANOVA, die komplexere, aber auch mächtigere Lösung ist eine Multi-Level-Regressionsanalyse.

Wie bei normaler ANOVA und Regression auch, kann die Multi-Level-Regression alles, was die Messwiederholungs-ANOVA kann, und noch vieles mehr. So kann die ANOVA zwar äquivalent eines Random-Intercept-Modells die individuell unterschiedlichen Baselinewerte der verschiedenen Personen noch berücksichtigen, jedoch berücksichtigt die ANOVA nicht die Reihenfolge der untersuchten Bedingungen/Zeitpunkte. Das hat zur Folge, dass es in der Messwiederholungs-ANOVA kein Äquivalent zu Random Slopes gibt. (Stört einen eingefleischten ANOVA-Fan vermutlich aber auch nicht, er hat sich ja auch schon mit der Ungerichtetheit seiner Ergebnisse abgefunden.)

In R gibt es natürlich die Möglichkeit, beide Herangehensweisen umzusetzen. Im Sinne der gesteigerten Einheitlichkeit mit der Vorlesung und den anderen Einheiten der Übung fokussieren wir uns aber auf den Multi-Level-Ansatz, also im Wesentlichen den `lme()`-Befehl. Für den ANOVA-Ansatz ist stattdessen vor allem der `ezANOVA()`-Befehl aus dem Package "ez" relevant. Der Field behandelt beide Lösungswege, präferiert aber ebenfalls den Multi-Level-Ansatz. Für Sie ist an dieser Stelle aber primär ein Bewusstsein dafür wichtig, dass es diese zwei Alternativen gibt, vor allem, wenn Sie im Internet oder im Field nach Hilfe suchen. Hier werden Sie häufig Code finden, der auf `ezANOVA()` beruht, und deshalb nur bedingt mit dem hier Gelernten übereinstimmt. Das ez-Package nutzt ausserdem eine leicht exotische Syntax, die zunächst verwirrend wirken kann.

Weitere Stolpersteine bei der Suche im Internet sind die Quadratsummen-Typen und die Sphärizitätsannahme. Die Quadratsummen-Typen bestimmen Sie bei Multi-Level-Modellen indirekt darüber, womit Sie Ihr jeweiliges Modell vergleichen (dem Nullmodell, einem Modell mit allen zu kontrollierenden Störvariablen, oder einem Modell mit allen Störvariablen und allen Interaktionen). Sphärizität wird bei Multi-Level-Modellen schlicht nicht angenommen, bzw. bei Random Slopes sogar explizit verletzt. Ganz allgemein steht Ihnen aber natürlich trotz des Multi-Level-Fokuses hier natürlich frei, alle angemessen damit lösbaren Aufgaben und Probleme auch mit jedem Package oder Befehl Ihrer Wahl zu lösen!

1) Messwiederholung

1. Daten einlesen Laden Sie das tidyverse und lesen Sie den Datensatz https://pzezula.pages.gwdg.de/data/text_messages.csv. Er enthält die Daten eines Experimentes, bei dem eine Gruppe für sechs Monate exzessiv ihr Mobiltelefon benutzen sollte, während der anderen Gruppe jede Nutzung ihres Mobiltelefons verboten wurde. Damit sollte der Einfluss von Kurznachrichten auf grammatikalische Fähigkeiten untersucht werden. Sowohl vor, als auch nach den sechs Monaten füllten sie deshalb einen standardisierten Grammatiktest aus, dessen Ergebnis in den beiden letzten Spalten vermerkt ist. Die Daten sind Tabstop-getrennt (`delim = "\t"`).

2. Daten transformieren Die Daten befinden sich aktuell im wide-Format. Transformieren Sie sie mit dem `gather()`-Befehl in das für R benötigte long-Format und speichern Sie sie als `text_messages`. Für den `gather()`-Befehl müssen Sie zunächst noch eine Spalte mit VP-Nummern von 1 bis 50 in den Datensatz einfügen. Faktorisieren Sie im Anschluss die Bedingungs- und Zeitvariable.

3. Plot Erzeugen Sie einen Linienplot mit Fehlerbalken, in dem die Entwicklung der grammatikalischen Fähigkeiten beider Gruppen dargestellt wird. Es zeigt sich, dass auch die Kontrollgruppe nach sechs Monaten schlechter im Grammatiktest abschneidet als zur Baseline. Überlegen Sie sich mögliche Gründe. Nutzen Sie die von Ihnen gesetzten Fehlerbalken, um abzuschätzen, ob die Verschlechterung der Kontrollgruppe signifikant sein könnte.

4. Signifikanztest Prüfen Sie, ob sich die grammatikalischen Fähigkeiten der Handynutzer signifikant stärker verschlechterten als die der Nicht-Handynutzer. Durch das Mixed Design mit Messzeitpunkt als within-subject Variable sind Sie in der Lage, die Handy-unabhängigen Unterschiede in den Grammatik-Fähigkeiten der Probanden herauszurechnen. Hierzu weisen Sie jedem Probanden seinen eigenen Random Intercept zu, der quasi der Fähigkeit zu Beginn des Experiments entspricht.

Im `lme()`-Befehl tun Sie das, indem Sie als zusätzliches Argument `random = ~1 | participant` definieren. Hierdurch erhält jeder Proband einen eigenen konstanten (deswegen 1) Einfluss auf das Kriterium, der aus der Residualvarianz herausgerechnet wird. Bei gleicher durch die Prädiktoren erklärter Varianz steigt so der Wert der F-Statistik. Um diesen Effekt näher zu beleuchten, führen Sie im Anschluss eine weitere Analyse durch, bei der Sie die Messwiederholungsstruktur der Daten ignorieren, und mittels `lm()` oder besser mit `nlme::gls()` ein zweites Modell mit den gleichen Prädiktoren, aber ohne Random Intercepts erstellen. Wie fällt jetzt die Untersuchung des Interaktionsterms aus?

Lösung

```
library(tidyverse)
# text_messages_wide <- read_delim("https://pzezula.pages.gwdg.de/data/text_messages.dat", "\t")
text_messages_wide <- readr::read_delim("http://md.psych.bio.uni-goettingen.de/mv/data/div/text_messages")
```

Unteraufgabe 1

```
##
## -- Column specification -----
## cols(
##   Group = col_character(),
##   Baseline = col_double(),
##   Six_months = col_double()
## )
```

```
text_messages_wide$participant <- c(1:nrow(text_messages_wide))
text_messages <- gather(text_messages_wide, time, grammar, Baseline:Six_months)
text_messages %>% mutate(group_fac = factor(Group)) %>%
  mutate(time_fac = factor(time)) -> text_messages
```

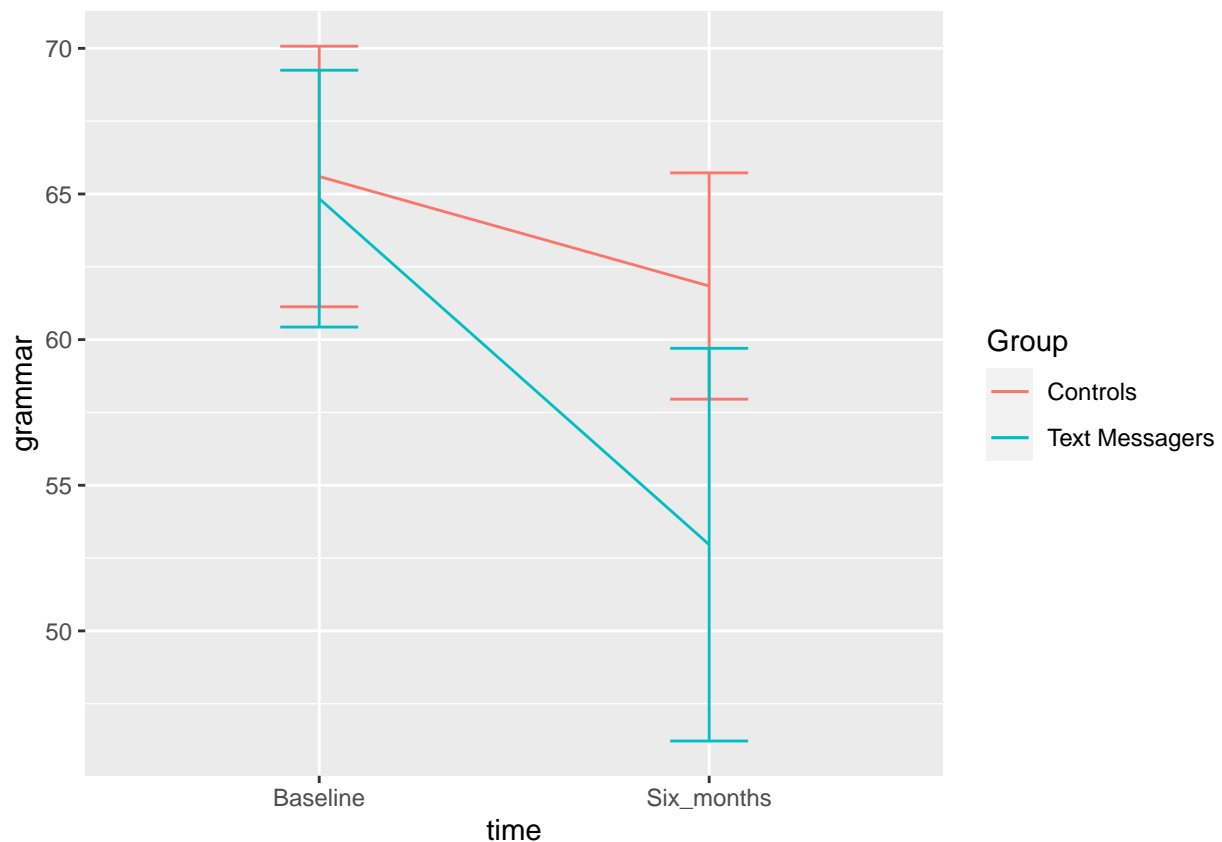
Unteraufgabe 2

```
text_messages_plot <- ggplot(data = text_messages, aes(
  x = time,
  y = grammar,
  group = Group,
  color = Group
))

text_messages_plot + stat_summary(fun.y = mean, geom = "line") +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar", width = .2)
```

Unteraufgabe 3

Warning: `fun.y` is deprecated. Use `fun` instead.



Möglicherweise hielt sich die Kontrollgruppe nicht an das Handyverbot. Es könnte aber auch schlicht ein Motivationsabfall zwischen den beiden Zeitpunkten geben, weil das Teilnehmen an einer Studie den Reiz des Neuen verliert. Vielleicht werden Erwachsene nach Ende der Schulzeit auch schlicht immer kontinuierlich schlechter in Grammatik. Die hier verwendeten Fehlerbalken lassen allerdings keine signifikante Verschlechterung der Kontrollgruppe vermuten: Das 95%-Konfidenzintervall des einen Wertes schliesst den anderen Wert mit ein. Wenn Sie stattdessen Standardfehler als Fehlerbalken genutzt haben, erkennen Sie die gleiche Information daran, dass sich die Fehlerbalken überlappen.

```
library(nlme)
text_messages_model <- lme(grammar ~ group_fac * time_fac,
                           data = text_messages,
                           random = ~1 | participant, method="ML")
summary(text_messages_model)
```

Unteraufgabe 4

```
## Linear mixed-effects model fit by maximum likelihood
## Data: text_messages
##      AIC      BIC    logLik
##  784.9265 800.5575 -386.4633
##
## Random effects:
## Formula: ~1 | participant
##      (Intercept) Residual
## StdDev:      6.772358 9.744331
##
## Fixed effects: grammar ~ group_fac * time_fac
##
##              Value Std.Error DF   t-value p-value
## (Intercept)    65.60  2.422265 48 27.082092  0.0000
## group_facText Messagers    -0.76  3.425600 48 -0.221859  0.8254
## time_facSix_months    -3.76  2.812946 48 -1.336677  0.1876
## group_facText Messagers:time_facSix_months -8.12  3.978106 48 -2.041172  0.0468
## Correlation:
##
##              (Intr) grp_TM tm_fS_
## group_facText Messagers    -0.707
## time_facSix_months    -0.581  0.411
## group_facText Messagers:time_facSix_months  0.411 -0.581 -0.707
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -3.75999093 -0.46824711  0.02633618  0.48669764  2.20450296
##
## Number of Observations: 100
## Number of Groups: 50
```

#Interaktionsterm (knapp) signifikant

```
text_messages_lm <- lm(grammar ~ group_fac * time_fac, data = text_messages)
summary(text_messages_lm)
```

```
##
## Call:
## lm(formula = grammar ~ group_fac * time_fac, data = text_messages)
##
## Residuals:
##      Min      1Q  Median      3Q      Max
## -43.96  -6.84   0.16   8.07  25.04
##
## Coefficients:
```

```
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    65.600      2.422  27.082   <2e-16 ***
## group_facText Messengers        -0.760      3.426  -0.222    0.825
## time_facSix_months             -3.760      3.426  -1.098    0.275
## group_facText Messengers:time_facSix_months -8.120      4.845  -1.676    0.097 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.11 on 96 degrees of freedom
## Multiple R-squared:  0.1519, Adjusted R-squared:  0.1254
## F-statistic:  5.73 on 3 and 96 DF,  p-value: 0.001192

#Interaktionsterm nicht mehr signifikant

# mit gls()
text_messages_gls <- nlme::gls(grammar ~ group_fac * time_fac, data = text_messages, method="ML")
summary(text_messages_gls)

## Generalized least squares fit by maximum likelihood
## Model: grammar ~ group_fac * time_fac
## Data: text_messages
##      AIC      BIC    logLik
## 788.5337 801.5595 -389.2668
##
## Coefficients:
##                                Value Std.Error   t-value p-value
## (Intercept)                    65.60  2.422265  27.082093  0.0000
## group_facText Messengers        -0.76  3.425600  -0.221859  0.8249
## time_facSix_months             -3.76  3.425600  -1.097618  0.2751
## group_facText Messengers:time_facSix_months -8.12  4.844530  -1.676117  0.0970
##
## Correlation:
##                                (Intr) grp_TM tm_fS_
## group_facText Messengers        -0.707
## time_facSix_months             -0.707  0.500
## group_facText Messengers:time_facSix_months  0.500 -0.707 -0.707
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -3.70450724 -0.57640650  0.01348319  0.68005854  2.11011969
##
## Residual standard error: 11.86663
## Degrees of freedom: 100 total; 96 residual
```

2) Zeitreihen

1. Daten einlesen Lesen Sie den Datensatz <https://pzezula.pages.gwdg.de/data/gdp.dat> als `gdp_wide` in R ein. In ihm finden Sie das Pro-Kopf-Bruttoinlandsprodukt (GDP per capita) neun verschiedener Länder zwischen den Jahren 1950 und 1983. Die Daten sind Tabstopp-getrennt (`delim = "\t"`).

2. Daten transformieren Die Daten befinden sich aktuell im wide-Format. Transformieren Sie sie mit dem `gather()` in das für R benötigte long-Format und speichern Sie sie als `gdp`. Faktorisieren Sie im

Anschluss die Jahres- und Ländervariable.

3. Explorativer Plot Erstellen Sie einen Line-Plot, um die Entwicklung der einzelnen Länder über die Jahre hinweg darzustellen. Nutzen Sie die `color-aesthetic`, um die Länder zu kodieren. Speichern Sie den vollständigen Plot als Objekt ab, damit wir ihn in einer späteren Aufgabe erweitern können.

4. Modellierung Erstellen Sie drei mögliche Modelle für die GDP-Entwicklung über die Jahre: `m1` ohne Random Effects (`lm()`), nur mit den Jahren als Prädiktor, `m2` mit Random Intercepts für die einzelnen Länder, und `m3` mit Random Intercepts und Random Slopes. Die Syntax für Random Slopes in `lme()` lautet `random = ~praediktor_dessen_slope_random_soll | gruppe`.

Verwenden Sie `method = "ML"`, damit die Modelle miteinander verglichen werden können. Vergleichen Sie mittels ANOVA `m2` mit `m3`. Welches Modell erklärt die Daten besser? Wichtiger Hinweis: Da die Jahreszahl selbst Schwierigkeiten beim Anpassen des Modells macht, ist es besser, mit der laufenden Nummer des Jahres zu arbeiten. So wird in Variable `year_nr` 1950 -> 1, 1951 -> 2 usw. In den Abbildungen passen wir die Tick-Labels entsprechend an, so dass die Jahreszahlen wieder erscheinen.

5. Modelle plotten Speichern Sie die Vorhersagen der drei Modelle mit z.B. `pred_m1 <- predict(m1)` in jeweils ein eigenes Objekt. Erzeugen Sie für jedes Modell einen Plot, indem Sie dem Plot aus Unteraufgabe 3 ein neues `geom_line()` hinzufügen, dessen `y-aesthetic` Sie zu der eben erzeugten Vorhersagenvariable des jeweiligen Modells ändern.

Lösung

```
library(tidyverse)
# gdp_wide <- read_delim("https://pzezula.pages.gwdg.de/data/gdp.dat", "\t")
gdp_wide <- read_delim("http://md.psych.bio.uni-goettingen.de/mv/data/div/gdp.dat", "\t")
```

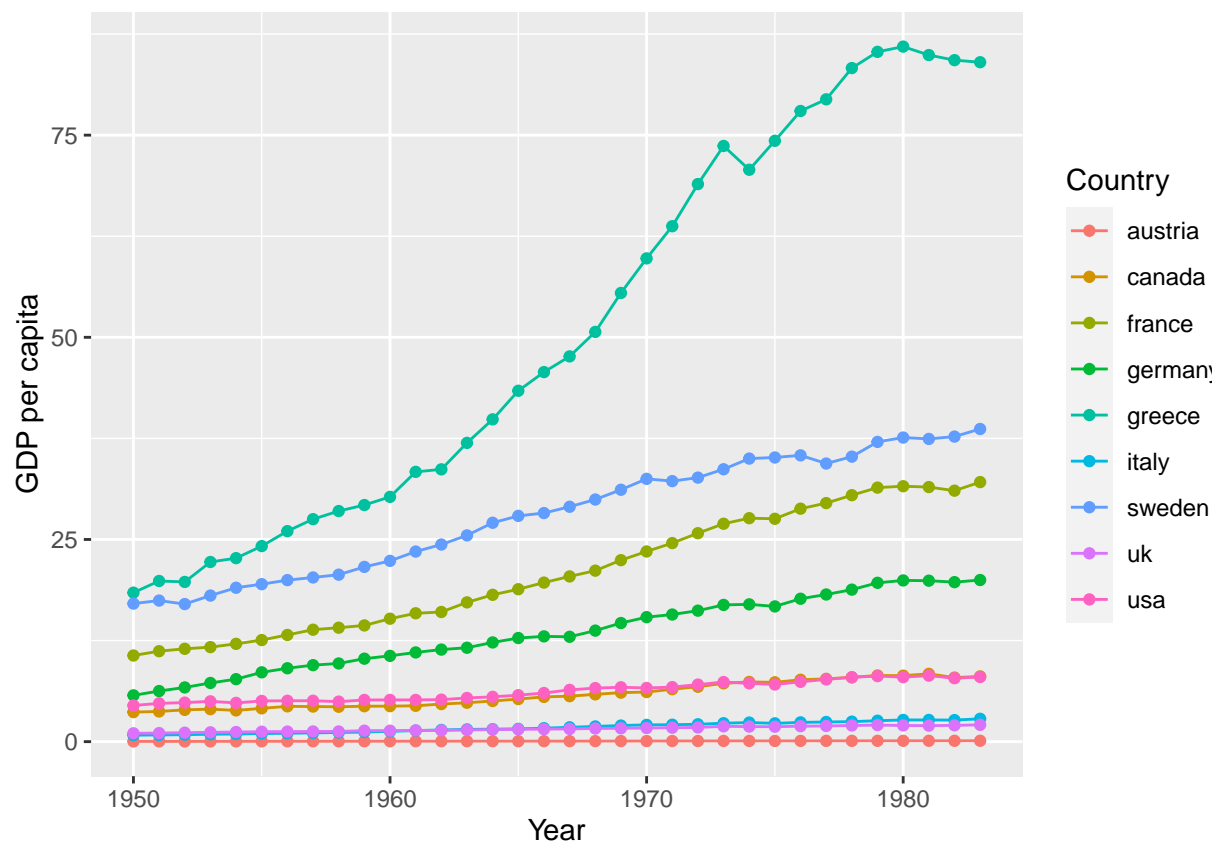
Unteraufgabe 1

```
##
## -- Column specification -----
## cols(
##   year = col_double(),
##   austria = col_double(),
##   canada = col_double(),
##   france = col_double(),
##   germany = col_double(),
##   greece = col_double(),
##   italy = col_double(),
##   sweden = col_double(),
##   uk = col_double(),
##   usa = col_double()
## )
```

```
gdp <- gather(gdp_wide, country, gdp, austria:usa)
gdp %>% mutate(country_fac = factor(country)) %>%
  mutate(year_fac = factor(year)) %>%
  mutate(year_nr = year - 1949) -> gdp
```

Unteraufgabe 2

```
(gdp_plot_1 <- ggplot(data = gdp, aes(year_nr, gdp,
                                     group = country_fac,
                                     color = country_fac)) +
  geom_line() +
  geom_point() +
  scale_x_continuous(breaks = c(1, 11, 21, 31), label = c("1950", "1960", "1970", "1980")) +
  labs(x = "Year", y = "GDP per capita", color = "Country")
)
```



Unteraufgabe 3

```
library(nlme)
# m1 <- lm(gdp ~ year, data = gdp)
m1 <- lm(gdp ~ year_nr, data = gdp)
```



```

m2 <- lme(gdp ~ year_nr, data = gdp,
          random = ~1 | country_fac,
          method = "ML")

m3 <- lme(gdp ~ year_nr, data = gdp,
          random = ~1 + year_nr | country_fac,
          method = "ML")

summary(m1)

```

Unteraufgabe 4

```

##
## Call:
## lm(formula = gdp ~ year_nr, data = gdp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.777 -11.579  -4.951   9.115  64.622
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.2293     2.0305   2.575  0.0105 *
## year_nr       0.5193     0.1012   5.131 5.15e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.37 on 304 degrees of freedom
## Multiple R-squared:  0.0797, Adjusted R-squared:  0.07667
## F-statistic: 26.33 on 1 and 304 DF, p-value: 5.152e-07

```

```
summary(m2)
```

```

## Linear mixed-effects model fit by maximum likelihood
##   Data: gdp
##       AIC      BIC    logLik
##  2136.28 2151.175 -1064.14
##
## Random effects:
##   Formula: ~1 | country_fac
##      (Intercept) Residual
## StdDev:    15.71083 7.272041
##
## Fixed effects:  gdp ~ year_nr
##              Value Std.Error  DF   t-value p-value
## (Intercept) 5.229306  5.322917 296   0.982414  0.3267
## year_nr      0.519281  0.042513 296  12.214712  0.0000
## Correlation:
##      (Intr)
## year_nr -0.14
##
## Standardized Within-Group Residuals:

```

```
##           Min           Q1           Med           Q3           Max
## -3.260696450 -0.361469341 -0.008045481  0.375195313  3.882753955
##
## Number of Observations: 306
## Number of Groups: 9
```

```
summary(m3)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: gdp
##       AIC      BIC    logLik
## 1234.862 1257.203 -611.431
##
## Random effects:
## Formula: ~1 + year_nr | country_fac
## Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev    Corr
## (Intercept) 4.6909664 (Intr)
## year_nr      0.7146444 0.596
## Residual     1.4955014
##
## Fixed effects: gdp ~ year_nr
##           Value Std.Error DF t-value p-value
## (Intercept) 5.229306  1.578566 296 3.312694  0.0010
## year_nr      0.519281  0.239157 296 2.171298  0.0307
## Correlation:
##           (Intr)
## year_nr 0.588
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -4.368512622 -0.123322614 -0.008279835  0.134552897  4.812089353
##
## Number of Observations: 306
## Number of Groups: 9
```

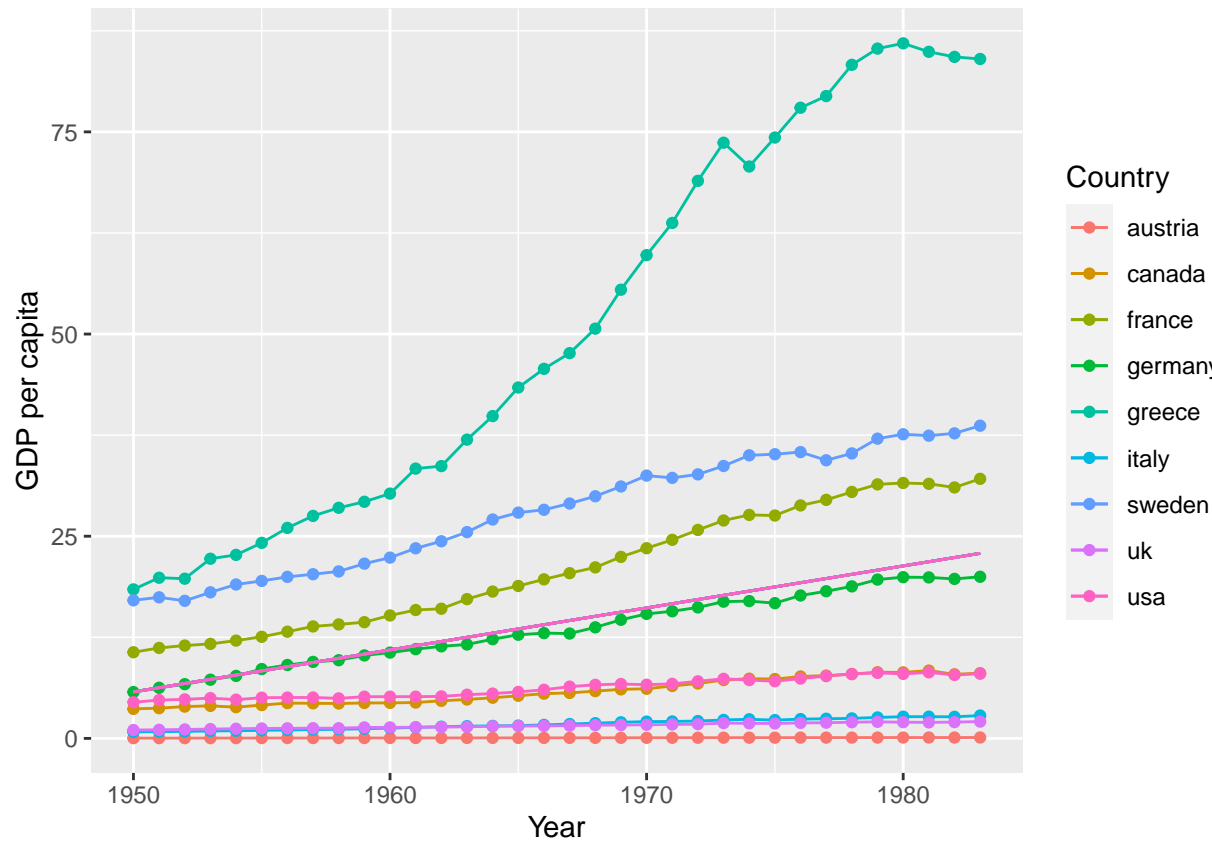
```
anova(m2, m3)
```

```
##   Model df      AIC      BIC    logLik  Test  L.Ratio p-value
## m2     1  4 2136.280 2151.175 -1064.140
## m3     2  6 1234.862 1257.204  -611.431 1 vs 2 905.4185  <.0001
```

```
#m3 sehr viel besser
```

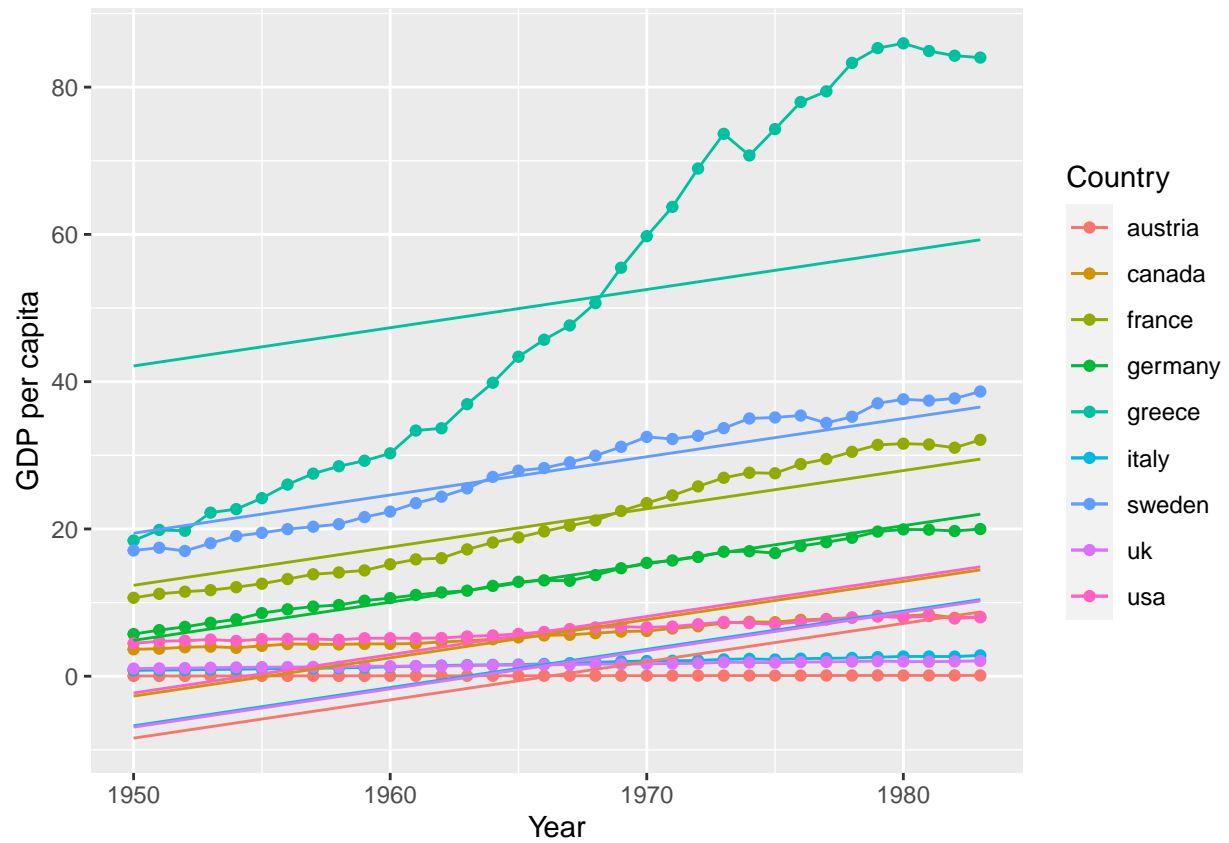
```
pred_m1 <- predict(m1)
pred_m2 <- predict(m2)
pred_m3 <- predict(m3)

( gdp_plot_m1 <- gdp_plot_1 + geom_line(aes(y = pred_m1)) )
```



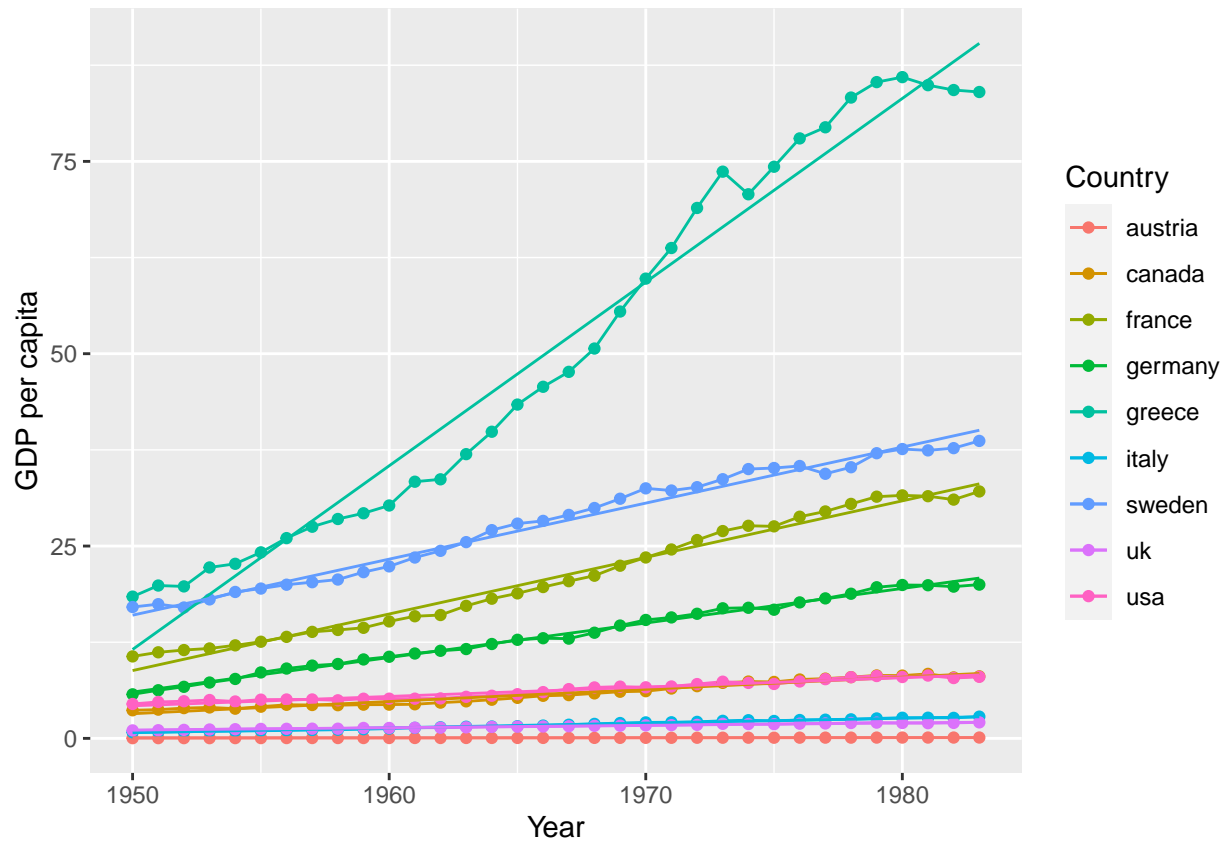
Unteraufgabe 5

```
# gdp_plot_m1
( gdp_plot_m2 <- gdp_plot_1 + geom_line(aes(y = pred_m2)) )
```



```
#gdp_plot_m2
```

```
( gdp_plot_m3 <- gdp_plot_1 + geom_line(aes(y = pred_m3)) )
```



```
#gdp_plot_m3
```

Literatur

Anmerkung: Diese Übungszettel basieren zum Teil auf Aufgaben aus dem Lehrbuch *Discovering Statistics Using R* (Field, Miles & Field, 2012). Sie wurden für den Zweck dieser Übung modifiziert, und der verwendete R-Code wurde aktualisiert.

Field, A., Miles, J., & Field, Z. (2012). *Discovering Statistics Using R*. London: SAGE Publications Ltd.

English

Links

[Exercise sheet as PDF](#)

Exercise sheet with solutions included

[Exercise sheet with solutions included as PDF](#)

[The source code of this sheet as .Rmd](#) (Right click and “save as” to download ...)

Some hints

1. Please try to solve this sheet in an .Rmd file. You can create one from scratch using **File > New file > R Markdown...**. You can delete the text beneath *Setup Chunk* (starting from line 11). Alternatively, you can download our template file under [this link](#) (right click > save as...).
2. You'll find a lot of the important information on the [website of this course](#)
3. Please don't hesitate to search the web for help with this sheet. In fact, being able to effectively search the web for problem solutions is a very useful skill, even R pros work this way all the time! The best starting point for this is the [R section on the programming site Stackoverflow](#)
4. On the R Studio website, you'll find highly helpful [cheat sheets](#) for many of R topics. The [base R cheat sheet](#) might be a good starting point.

Ressources

Since this is a hands-on seminar, we won't be able to present each and every new command to you explicitly. Instead, you'll find here references to helpful ressources that you can use for completing this sheets.

Ressource	Description
Field, chapter 13	This is about pure repeated measures ANOVA
Field, chapter 14	This chapter deals with mixed designs, within subjects and between subjects factors
Field, chapter 19	This chapter is about multi level analyses in general

Hint of the week

Efficient model formulae

If you want to specify a model using the command `lm()`, the fully written model can become large. We often have shortcuts. So in order to write `reaction_variable ~ predictor1 + predictor2 + predictor1:predictor2` to fit a two predictor with interaction model, we could also write `reaction_variable ~ predictor1 * predictor2` to specify the same model in R.

Some general remarks

This week we deal with situations, in which we measure the same person (observation) repeatedly under different conditions or in different situations or at different times. This usually violates the preconditions of regression, because repeated measures of the same person usually correlate and thus are not independent - we have autocorrelation. We usually solve this problem by applying repeated measures ANOVA. A more complex but also more powerful alternative would be multi level analysis.

Regression can do all, a "normal" ANOVA does. The same applies to Multi level analysis, that can do all, classical repeated measures ANOVA does, and a lot more.

A random intercept model would simulate an ANOVA. But there is no equivalent for a random slope model in the "ANOVA world". A random slope model could estimate the influence of repetition more adequately.

R offers us both possibilities. Due to the concept in this course however we focus on multi level analysis here. This means, that we mainly deal with the command `lme()`. Just to mention it: The "classical" repeated ANOVA can be done using the package `ez` using `ezANOVA()`. Field (2012) deals with both ways with preference to multi level analysis. You should be aware, that there are these two possibilities to deal with repeated measures designs. This is mainly due to the fact, that you will often find code snippets of `ezANOVA()` or others,

f. e. when searching the internet or studying Field (2012). Moreover `ezANOVA()` uses a slightly exotic syntax, that might confuse.

There are more pitfalls with this topic when searching the internet: “Types of sums of squares” (SS) and “sphericity”. In multi level analyses you sort of deal with types of SS indirectly, depending on which models you compare (null model or model with all variables to control or model with all interactions additionaly). In multi level analysis there simply is no precondition of sphericity.

Nevertheless, although we focus on the multilevel access here, there are, as often, a lot of possibilities to go. As long as there is a appropriate modeling solution in whatever package, you are free to use it.

1) Repeated measure designs

1. Read data

1. Load the necessary packages and set a reasonable working directory.
2. Load the dataframe from link https://pzezula.pages.gwdg.de/data/text_messages.dat in whatever way you like.

The data show two groups, one of which was asked to use their mobile phone excessively whereas the second group wasn't allowed to use their phone. The idea was, to examine the influence of short messages (this was in the days of SMS) on grammatical abilities. Subjects completed a standardized grammer test before and after the six months period. You find the resulting scores in the last two columns of the data. Data are tabulator delimited (`delim = "\t"`).

2. Data transformation The data are in wide format. Transform the data to the long format that R needs to analyze this kind of data. Use the command `gather()` to do that and store it to a data object called `text_messages`. Insert also a column with subject numbers 1 to 50 in our data table. Factorize the column that codes the condition and time after having converted the data.

3. Plot Create a line plot with error bars, where you visualize the development of the grammatical abilities of both groups. The results indicate, that our control group has worse results in the grammer test compared to the baseline. Think of reasons, why this might have happened. Use the error bars to get an idea, whether the difference in the control group could be significant.

4. Test of significance Test, whether the handy users get even significantly worse than the subjects without handy usage. By applying a mixed design with measurement time beeing a within subject variable you can correct the data for the differences in subjects, that do not depend on the (non-) usage of the mobile phone. To do that, you can get a random intercept for every subject, that is basicly the ability of the users at the start of the experiment.

You can do that by adding the argument `random = ~1 | participant` to the `lme()` command. Doing this we have a “constant” (therefore set to 1) influence on the outcome for every subject. This corrects and reduces our residual variance. As the percentage of variance, that is explained by our predictors, keeps the same this leads to an increase of the resulting F-value. To have a closer look to that, do a further analysis, where you ignore the repeated measures structure of our data. Do the same analysis using `lm()` or even better using `nlme::gls()` with the same predictors but without random intercepts. Now, what is our interaction like?

Solution

```
library(tidyverse)
text_messages_wide <- read_delim("https://pzezula.pages.gwdg.de/data/text_messages.dat", "\t")
```

Subtask 1

```
##
## -- Column specification -----
## cols(
##   Group = col_character(),
##   Baseline = col_double(),
##   Six_months = col_double()
## )
```

```
# or
text_messages_wide <- read_tsv("https://pzezula.pages.gwdg.de/data/text_messages.dat")
```

```
##
## -- Column specification -----
## cols(
##   Group = col_character(),
##   Baseline = col_double(),
##   Six_months = col_double()
## )
```

```
text_messages_wide$participant <- c(1:nrow(text_messages_wide))
text_messages <- gather(text_messages_wide, time, grammar, Baseline:Six_months)
text_messages %>% mutate(group_fac = factor(Group)) %>%
  mutate(time_fac = factor(time)) -> text_messages
```

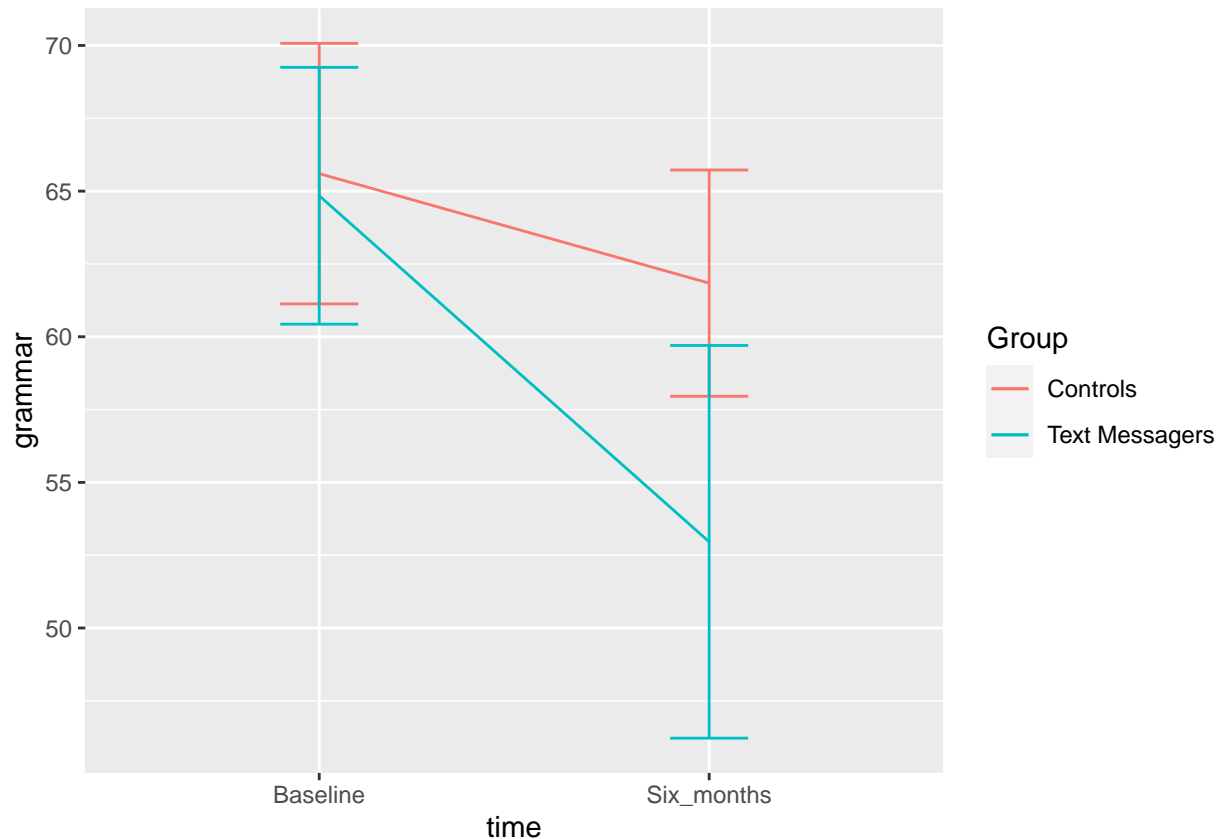
Subtask 2

```
text_messages_plot <- ggplot(data = text_messages, aes(
  x = time,
  y = grammar,
  group = Group,
  color = Group
))

text_messages_plot + stat_summary(fun.y = mean, geom = "line") +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar", width = .2)
```

Subtask 3

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

Our control group might not have respected our prohibition of mobile usage. Alternatively this could also be a drop in motivation in the time between the two measurements, because taking part in an experiment isn't that interesting any more after some time. Adults might even get worse in grammar after having finished school anyway.

The error bars do not indicate a significant drop of grammatical abilities in our control group. The 95% confidence interval indicated does not include the mean of the other condition. If you have used standard error instead, you have the same information, because the standard errors do not overlap.

```
library(nlme)
text_messages_model <- lme(grammar ~ Group*time,
                           data = text_messages,
                           random = ~1 | participant)
summary(text_messages_model)
```

Subtask 4

```
## Linear mixed-effects model fit by REML
## Data: text_messages
##      AIC      BIC    logLik
##  770.8039 786.1899 -379.4019
##
## Random effects:
## Formula: ~1 | participant
```

```
##          (Intercept) Residual
## StdDev:      6.912009 9.945266
##
## Fixed effects:  grammar ~ Group * time
##
##              Value Std.Error DF   t-value p-value
## (Intercept)      65.60  2.422265 48 27.082092  0.0000
## GroupText Messagers      -0.76  3.425600 48 -0.221859  0.8254
## timeSix_months      -3.76  2.812946 48 -1.336677  0.1876
## GroupText Messagers:timeSix_months -8.12  3.978106 48 -2.041172  0.0468
## Correlation:
##
##              (Intr) GrpTxM tmSx_m
## GroupText Messagers      -0.707
## timeSix_months      -0.581  0.411
## GroupText Messagers:timeSix_months  0.411 -0.581 -0.707
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -3.68402369 -0.45878658  0.02580408  0.47686435  2.15996300
##
## Number of Observations: 100
## Number of Groups: 50
```

#Interaktionsterm (knapp) signifikant

```
text_messages_lm <- lm(grammar ~ Group*time, data = text_messages)
summary(text_messages_lm)
```

```
##
## Call:
## lm(formula = grammar ~ Group * time, data = text_messages)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -43.96  -6.84   0.16   8.07  25.04
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      65.600      2.422  27.082  <2e-16 ***
## GroupText Messagers      -0.760      3.426  -0.222   0.825
## timeSix_months      -3.760      2.813  -1.098   0.275
## GroupText Messagers:timeSix_months -8.120      3.978  -1.676   0.097 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.11 on 96 degrees of freedom
## Multiple R-squared:  0.1519, Adjusted R-squared:  0.1254
## F-statistic:  5.73 on 3 and 96 DF,  p-value: 0.001192
```

#Interaktionsterm nicht mehr signifikant

2) Time series

1. Read data Read the data you can find in <https://pzezula.pages.gwdg.de/data/gdp.dat> in R and name it `gdp_wide`. You can find the GDP (Gross Domestic Product) per capita for 9 different countries from 1950 to 1983. The data format is tabulator delimited (`delim = "\t"`).

2. Data transformation Data are in wide format. Convert it using `gather()` to the long format that R needs for this type of data structure. Name the data object `gdp`. After conversion factorize the columns for year and country.

3. Explorative plot Create a line plot to show the development of every country over in the course of the years. Use the aesthetic color to code the countries. Store the complete plot as an object in order to extend it afterwards.

4. Modeling Fit three different models to test the GDP development:

- `m1` without random effects (use `lm()`) only year is predictor
- `m2` with random intercept for country additionally
- `m3` with random intercept and random slope To get random slopes with `lme()` use the syntax: The syntax to use to get random slopes is `random = ~ predictor_with_random_slope | group`.

Use `method = "ML"` to be able to compare the models. Compare `m2` and `m3` via `anova()`. Which model fits the data better? Important: Don't use year as is. Use `year_nr` instead. `year_nr <- year - 1949`. So year 1950 is `year_nr` 1 etc. We adapt tick labels on the x axis to refer back to the year. This is done because starting with 1950 causes problems when the model is adapted. <- Important: Don't use year as is. Use `as.numeric(faktorized_year)` instead. You may think this is unnecessary back and forth, but this causes that year starts with 1 and not with 1950. This is, how the maximum likelihood does the correct parameter estimation. ->

5. Plot the models Store the predicted values of the three models in an own object, f. e. `pred_m1 <- predict(m1)`. Create a plot for every model. Do this adding a new `geom_line()` to the saved base plot from above. You basically define a new y-aesthetic with the newly calculated predicted values.

Solution

```
library(tidyverse)
# gdp_wide <- read_delim("https://pzezula.pages.gwdg.de/data/gdp.dat", "\t")
gdp_wide <- read_delim("http://md.psych.bio.uni-goettingen.de/mv/data/div/gdp.dat", "\t")
```

Subtask 1

```
##
```

```
## -- Column specification -----
```

```
## cols(
```

```
##   year = col_double(),
##   austria = col_double(),
##   canada = col_double(),
##   france = col_double(),
##   germany = col_double(),
##   greece = col_double(),
##   italy = col_double(),
##   sweden = col_double(),
##   uk = col_double(),
##   usa = col_double()
## )
```

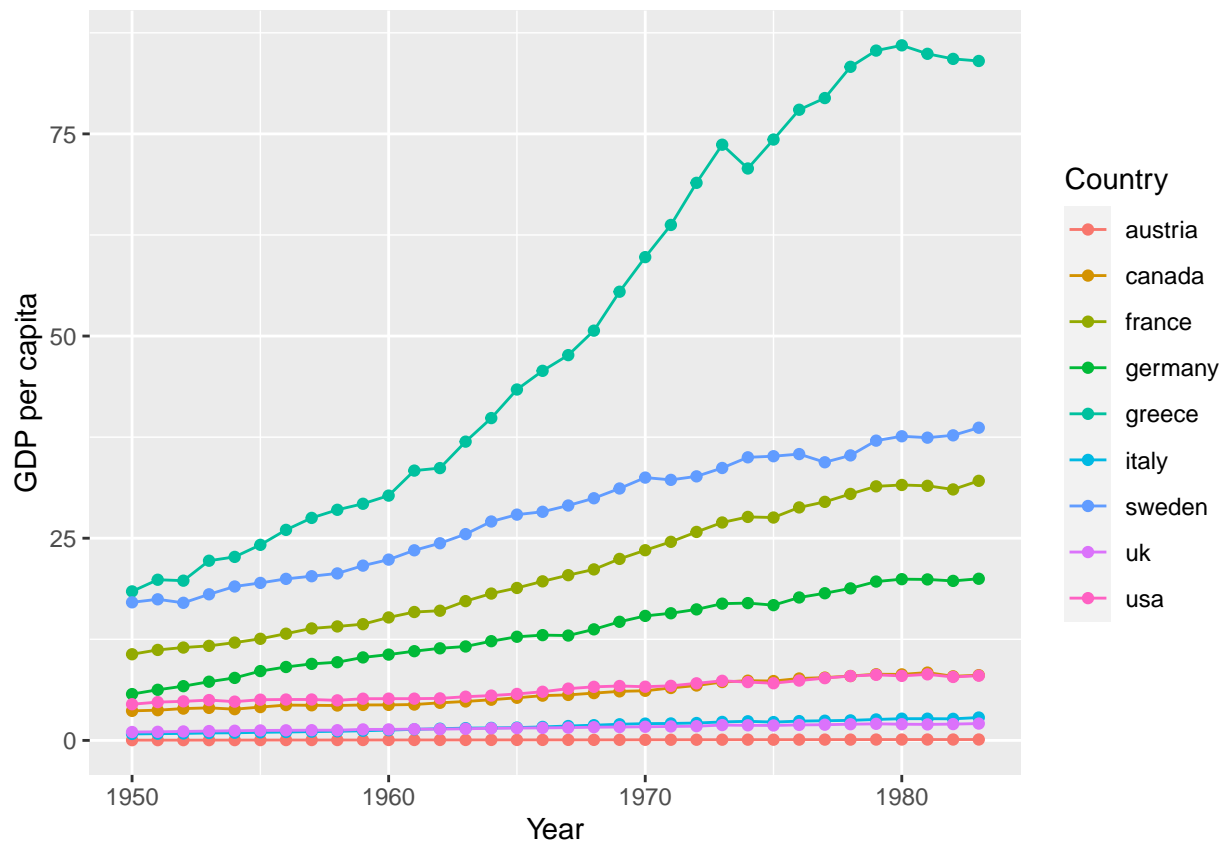
```
gdp <- gather(gdp_wide, country, gdp, austria:usa)
gdp %>% mutate(country_fac = factor(country)) %>%
  mutate(year_fac = factor(year)) %>%
  mutate(year_nr = year - 1949) -> gdp

gdp %>% mutate(country_fac = factor(country)) %>%
  mutate(year_fac = factor(year)) %>%
  mutate(year_nr = year - 1949) -> gdp
```

Subtask 2

```
gdp_plot_1 <- ggplot(data = gdp, aes(year, gdp,
                                     group = country_fac,
                                     color = country_fac)) +
  geom_line() +
  geom_point() +
  labs(x = "Year", y = "GDP per capita", color = "Country")

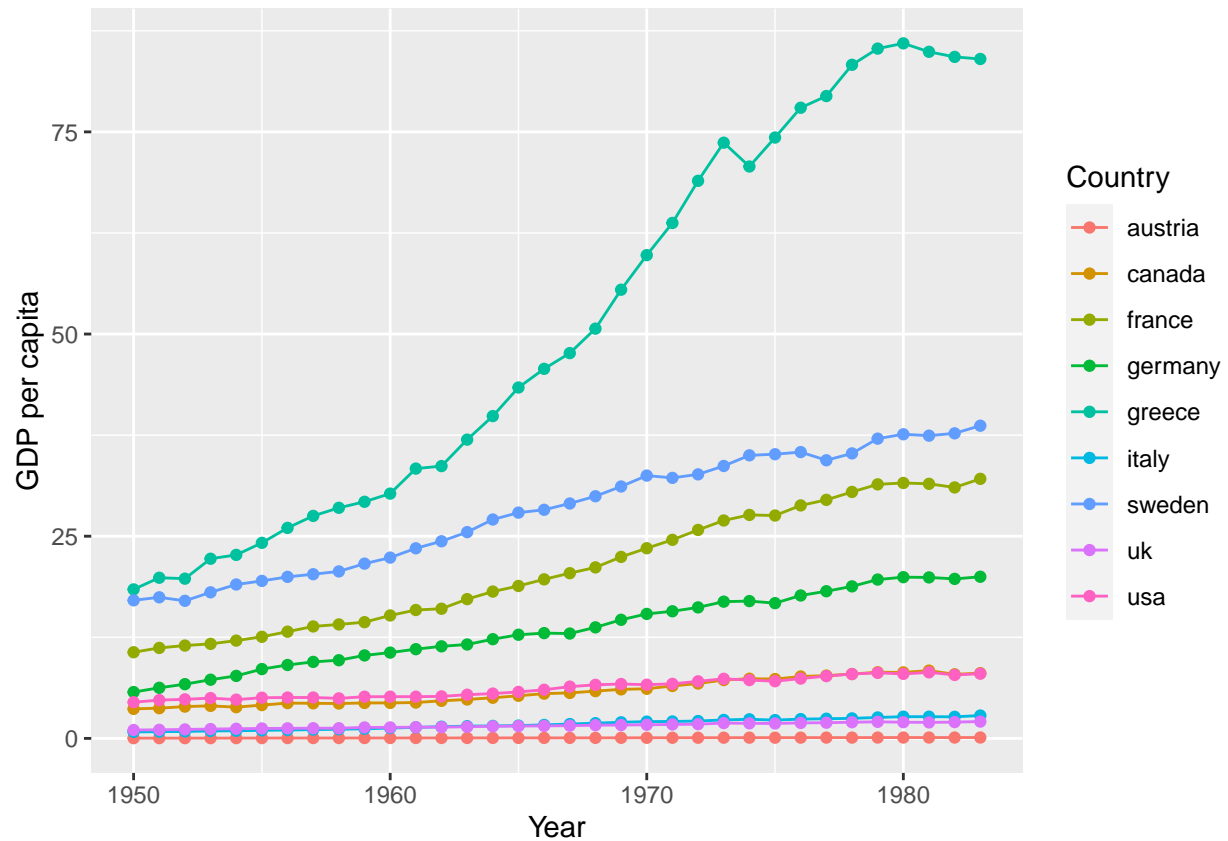
gdp_plot_1
```



Subtask 3

```
(gdp_plot_1 <- ggplot(data = gdp, aes(year_nr, gdp,
                                         group = country_fac,
                                         color = country_fac)) +

  geom_line() +
  geom_point() +
  scale_x_continuous(breaks = c(1, 11, 21, 31), label = c("1950", "1960", "1970", "1980")) +
  labs(x = "Year", y = "GDP per capita", color = "Country")
)
```



```
library(nlme)
m1 <- lm(gdp ~ year_nr, data = gdp)

m2 <- lme(gdp ~ year_nr, data = gdp,
          random = ~1 | country_fac,
          method = "ML")

m3 <- lme(gdp ~ year_nr, data = gdp,
          random = ~1 + year_nr | country_fac,
          method = "ML")

summary(m1)
```

Subtask 4

```
##
## Call:
## lm(formula = gdp ~ year_nr, data = gdp)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
##	-22.777	-11.579	-4.951	9.115	64.622

```
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.2293     2.0305   2.575  0.0105 *
## year_nr      0.5193     0.1012   5.131 5.15e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.37 on 304 degrees of freedom
## Multiple R-squared:  0.0797, Adjusted R-squared:  0.07667
## F-statistic: 26.33 on 1 and 304 DF, p-value: 5.152e-07
```

```
summary(m2)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: gdp
##       AIC      BIC    logLik
## 2136.28 2151.175 -1064.14
##
## Random effects:
## Formula: ~1 | country_fac
##      (Intercept) Residual
## StdDev:    15.71083  7.272041
##
## Fixed effects:  gdp ~ year_nr
##              Value Std.Error   DF   t-value p-value
## (Intercept) 5.229306  5.322917  296   0.982414  0.3267
## year_nr      0.519281  0.042513  296  12.214712  0.0000
## Correlation:
##      (Intr)
## year_nr -0.14
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -3.260696450 -0.361469341 -0.008045481  0.375195313  3.882753955
##
## Number of Observations: 306
## Number of Groups: 9
```

```
summary(m3)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: gdp
##       AIC      BIC    logLik
## 1234.862 1257.203 -611.431
##
## Random effects:
## Formula: ~1 + year_nr | country_fac
## Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev   Corr
## (Intercept) 4.6909664 (Intr)
## year_nr      0.7146444 0.596
## Residual     1.4955014
##
```

```
## Fixed effects: gdp ~ year_nr
##               Value Std.Error   DF  t-value p-value
## (Intercept)  5.229306   1.578566  296  3.312694   0.0010
## year_nr      0.519281   0.239157  296  2.171298   0.0307
## Correlation:
##      (Intr)
## year_nr 0.588
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -4.368512622 -0.123322614 -0.008279835  0.134552897  4.812089353
##
## Number of Observations: 306
## Number of Groups: 9
```

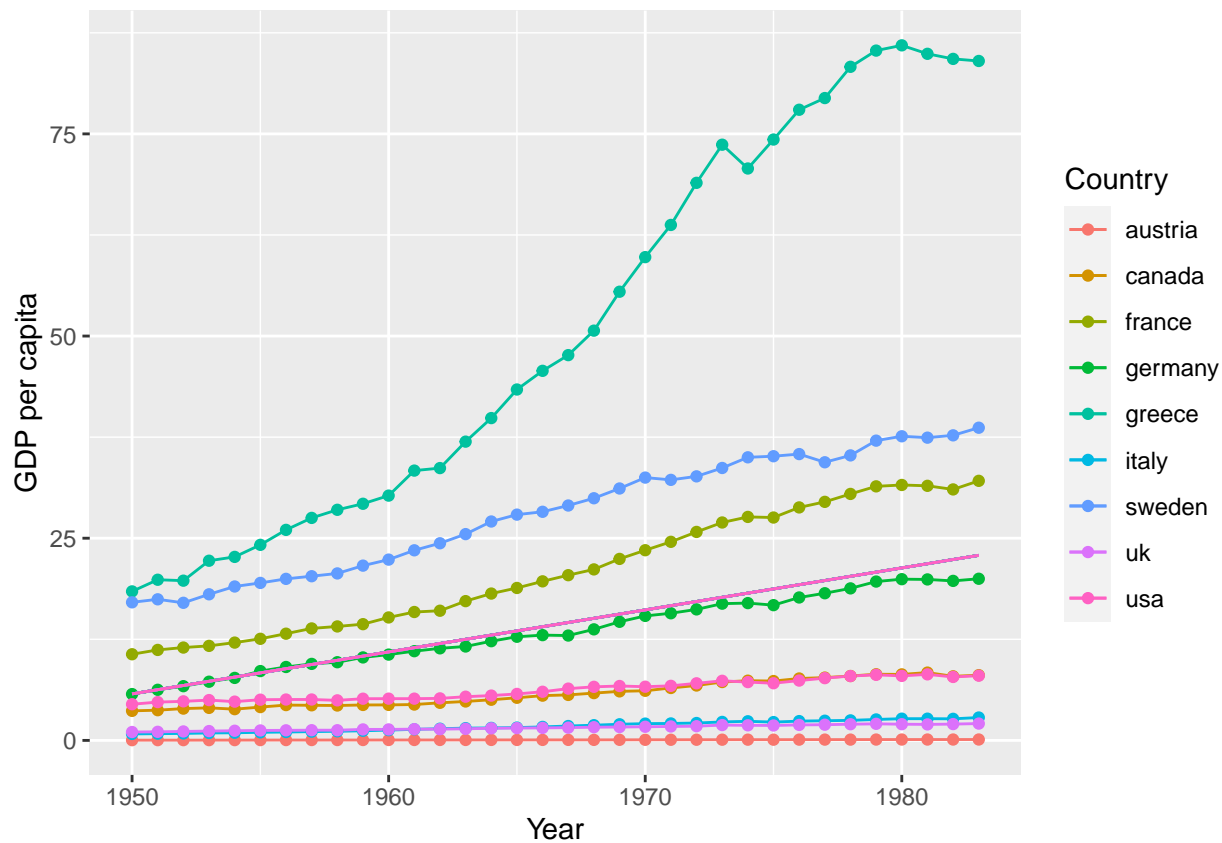
```
anova(m2, m3)
```

```
##      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## m2      1  4 2136.280 2151.175 -1064.140
## m3      2  6 1234.862 1257.204  -611.431 1 vs 2  905.4185  <.0001
```

```
# m3 a lot better
```

```
pred_m1 <- predict(m1)
pred_m2 <- predict(m2)
pred_m3 <- predict(m3)

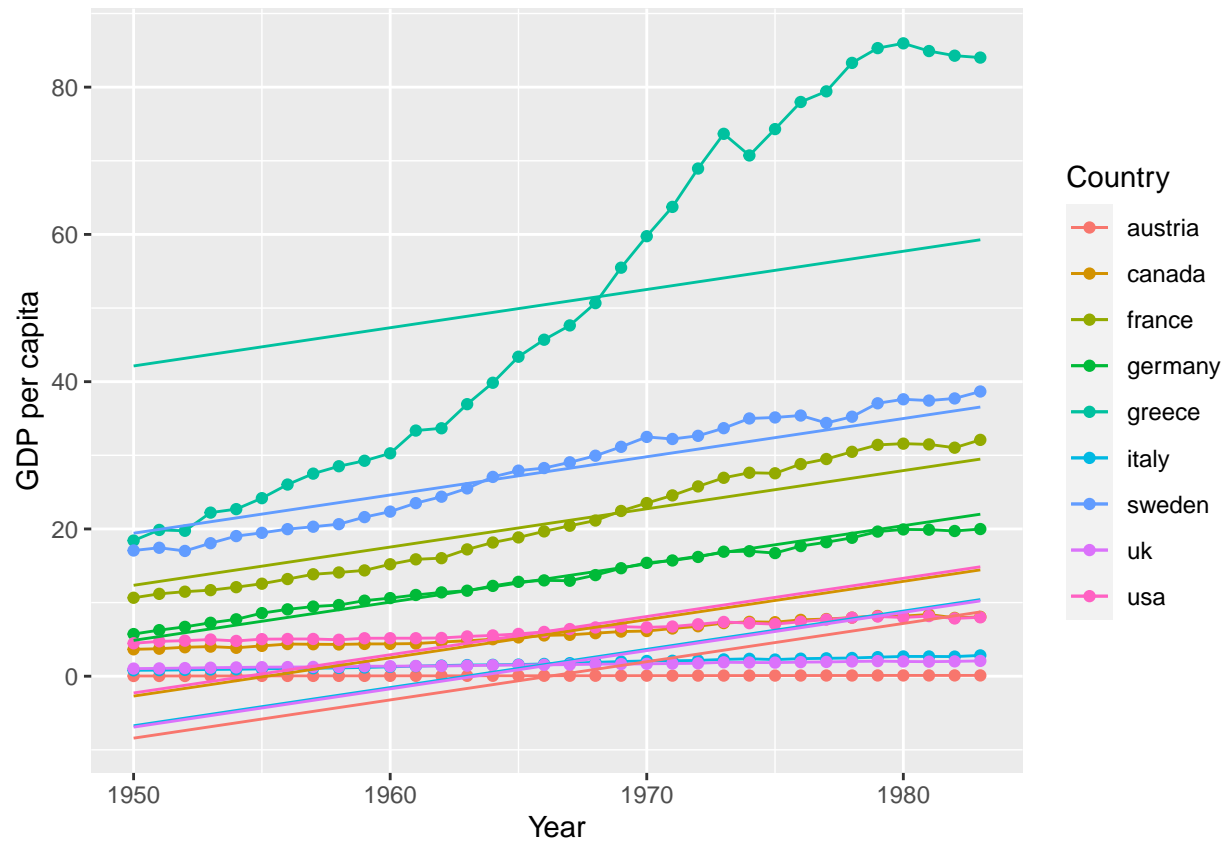
( gdp_plot_m1 <- gdp_plot_1 + geom_line(aes(y = pred_m1)) )
```

Subtask 5

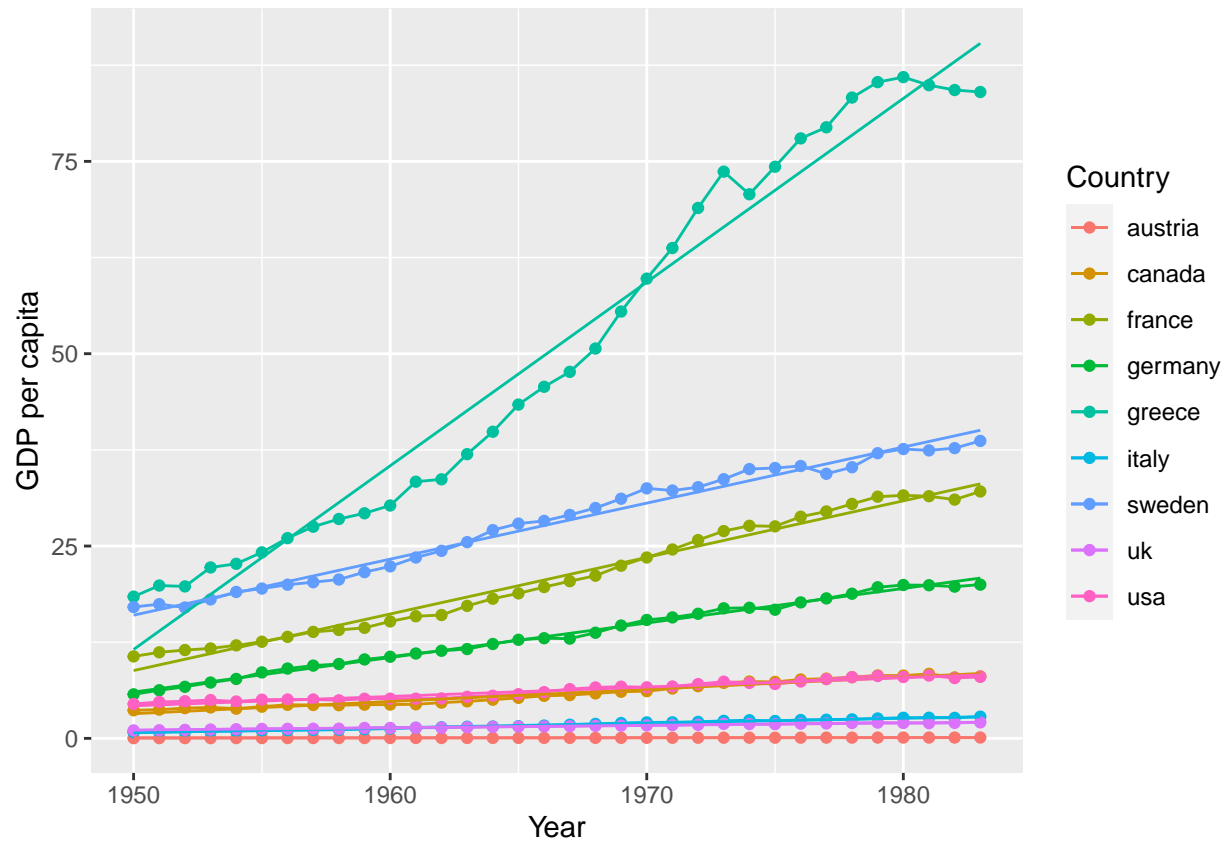
```
# gdp_plot_m1
```

```
( gdp_plot_m2 <- gdp_plot_1 + geom_line(aes(y = pred_m2)) )
```



```
#gdp_plot_m2
```

```
( gdp_plot_m3 <- gdp_plot_1 + geom_line(aes(y = pred_m3)) )
```



```
#gdp_plot_m3
```

Literature

Note: These sheets are based partially on exercises from the book *Discovering Statistics Using R* (Field, Miles & Field, 2012). They've been modified for the purposes of this seminar, and the R code was updated.

Field, A., Miles, J., & Field, Z. (2012). *Discovering Statistics Using R*. London: SAGE Publications Ltd.