

Prädiktoren der Lebenserwartung

Sebastian Sauer

Contents

Präambel	1
Forschungsfrage	1
Daten	1
Studiendesign	2
Datenanalyse	2
Setup	2
Daten importieren und aufbereiten	2
Importieren	2
Kontinente hinzufügen	2
Was ist die mittlere Lebenserwartung pro Kontinent im Jahr 2015 (laut den Daten)?	5
Extremwerte	8
Gibt es einen Zusammenhang von Lebenserwartung und dem Jahr?	16
Geovisualisierung	21
Weltkarte zeichnen	21
Karte einfärben (Choropleth)	23
Auf der Suche nach den USA	25

Präambel

Forschungsfrage

Die Lebenserwartung der Menschen variiert, wie jeder weiß. Die Variation wird u.a. erklärt durch das Land, in dem jemand lebt, aber auch durch die Zeit – in den letzten Jahrzehnten ist die Lebenserwartung zum Glück deutlich gestiegen. Allerdings nicht in jedem Jahr und in jedem Land in gleichem Maße. In dieser Fallstudie soll der Einfluss von Land, Region und Zeit auf die Lebenserwartung untersucht werden. Konkret soll die Lebenserwartung im Alter von 10 Jahren modelliert werden.

Daten

Die Daten stammen von Our World in Data, genauer gesagt aus dieser Analyse. Weitere Quellen sind hier dokumentiert.

Studiendesign

Es handelt sich um eine retrospektive Beobachtungsstudie. Die Daten wurden ex post erhoben.

Datenanalyse

Setup

Zuerst `mosaic` laden; nicht vergessen! Falls eines der Pakete auf Ihrer Maschine nicht installiert ist, bitte mit `install.packages("name_des_pakets")` installieren.

```
library(mosaic)
library(countrycode)
library(rnaturalearth)
library(tidyverse)
```

Daten importieren und aufbereiten

Importieren

Gehen wir davon aus, dass die Daten im Arbeitsverzeichnis liegen:

```
exp_raw <- read.csv("life-expectancy-at-age-10.csv")
head(exp_raw)
```

```
##      Entity Code Year e10..years.
## 1 Afghanistan  AFG 1950      39.58
## 2 Afghanistan  AFG 1955      41.08
## 3 Afghanistan  AFG 1960      42.41
## 4 Afghanistan  AFG 1965      43.68
## 5 Afghanistan  AFG 1970      44.98
## 6 Afghanistan  AFG 1975      46.44
```

Ich habe die Tabelle `exp_raw` genannt, um zu verdeutlichen, dass es sich um die “rohen”, unbearbeiteten Daten handelt. Häufig muss man die Daten noch modifizieren, da bietet es sich an, klar anzuzeigen, was die unbehandelten Daten und was die behandelten Daten sind.

Die Spalte `e10..years.` fasst die Lebenserwartung zum Alter von 10 Jahren. Allerdings ist der Name der Spalte nicht so schön. Benennen wir die Spalte um:

```
exp <- exp_raw %>%
  rename(e10 = e10..years.) # neu = alt
```

Kontinente hinzufügen

Es wäre doch schön, für jedes Land zu wissen, zu welchem Kontinent (oder Weltregion) es gehört. Natürlich kann man das händisch hinzufügen (z.B. in Excel). Aber komfortabler ist es, wenn man solche schnöde Arbeit den Computer erledigen lässt. In R gibt es ein Paket namens `countrycode` (und eine Funktion mit gleichem

Namen), das die Arbeit für uns übernimmt. Hier bilden wir den Landesnamen auf das zugehörige Kontinent ab; alternativ hätten wir auch die Spalte `Code` verwenden können. Allerdings ist nicht ganz klar, um *welchen* Code es sich handelt (vielleicht ISO). Das zeigt, das ein gutes Codebuch zu jedem Datensatz gehört!

```
exp2 <- exp %>%
  mutate(continent = countrycode(sourcevar = Entity, # Spalte, in der das Land steht
                                origin = "country.name", # Eingabe: Name (nicht z.B. ISO-Code) des La
                                destination = "continent")) # gewünschte Ausgabe: Name des Kontinents
```

```
## Warning in countrycode(sourcevar = Entity, origin = "country.name", destination = "continent"): Some
```

Dazu haben wir eine Spalte mit `mutate()` angelegt, die den Namen des Kontinents für jedes Land fasst. R informiert uns, dass einige Länder nicht einem Kontinent zugeordnet werden konnten. Schauen wir uns das genauer an. Wie viele Fälle liefern fehlende Werte zurück?

```
exp2 %>%
  summarise(is_na_sum = is.na(continent) %>% sum())
```

```
##   is_na_sum
## 1         120
```

Schauen wir uns ein Beispiel näher an:

```
exp2 %>%
  filter(Entity == "Timor")
```

```
##   Entity Code Year   e10 continent
## 1   Timor  TLS 1950 40.43      <NA>
## 2   Timor  TLS 1955 41.90      <NA>
## 3   Timor  TLS 1960 43.35      <NA>
## 4   Timor  TLS 1965 44.80      <NA>
## 5   Timor  TLS 1970 46.19      <NA>
## 6   Timor  TLS 1975 41.14      <NA>
## 7   Timor  TLS 1980 46.19      <NA>
## 8   Timor  TLS 1985 49.85      <NA>
## 9   Timor  TLS 1990 51.91      <NA>
## 10  Timor  TLS 1995 55.36      <NA>
## 11  Timor  TLS 2000 57.80      <NA>
## 12  Timor  TLS 2005 61.24      <NA>
## 13  Timor  TLS 2010 61.99      <NA>
## 14  Timor  TLS 2015 62.91      <NA>
## 15  Timor  TLS 2020 63.73      <NA>
## 16  Timor  TLS 2025 64.43      <NA>
## 17  Timor  TLS 2030 65.08      <NA>
## 18  Timor  TLS 2035 65.72      <NA>
## 19  Timor  TLS 2040 66.34      <NA>
## 20  Timor  TLS 2045 66.91      <NA>
## 21  Timor  TLS 2050 67.55      <NA>
## 22  Timor  TLS 2055 68.15      <NA>
## 23  Timor  TLS 2060 68.77      <NA>
## 24  Timor  TLS 2065 69.40      <NA>
## 25  Timor  TLS 2070 70.01      <NA>
```

```
## 26 Timor TLS 2075 70.60 <NA>
## 27 Timor TLS 2080 71.33 <NA>
## 28 Timor TLS 2085 72.03 <NA>
## 29 Timor TLS 2090 72.72 <NA>
## 30 Timor TLS 2095 73.39 <NA>
```

Vielleicht klappt es mit der Übersetzung der Spalte Code besser? Gehen wir mal davon aus, dass es sich um ISO-3661-Codes handelt.

```
exp2 <- exp2 %>%
  mutate(continent2 = countrycode(sourcevar = Code, # Spalte, in der das Land steht
                                origin = "iso3c", # Eingabe: Name (nicht z.B. ISO-Code) des Landes
                                destination = "continent")) # gewünschte Ausgabe: Name des Kontinents
```

```
## Warning in countrycode(sourcevar = Code, origin = "iso3c", destination = "continent"): Some values w
```

```
exp2 %>%
  summarise(is_na_sum = is.na(continent2) %>% sum())
```

```
##      is_na_sum
## 1           60
```

Besser; 60 fehlende Werte. Aber welche Fälle sind übrig? Schauen wir uns die mal an:

```
exp2 %>%
  filter(is.na(continent2)) %>%
  distinct(Entity)
```

```
##      Entity
## 1 Channel Islands
## 2      World
```

Auf Deutsch übersetzt heißt die Syntax oben:

Hey R, nimm die Tabelle exp2 UND DANN
 filtere die Fälle, die keine Werte haben für continent2 UND DANN
 zeige alle verschiedenen Werte für Entity

Laut dieser Quelle gehören die Channel Islands zu UK (GB). Es ist vielleicht pragmatisch, dies einfach zu übernehmen.

```
exp3 <- exp2 %>%
  mutate(continent2 = as.character(continent2)) %>%
  mutate(continent2 = case_when(
    Entity == "Channel Islands" ~ "Europe",
    Entity == "World" ~ "World",
    TRUE ~ continent2))
```

Prüfen wir, ob jetzt alle fehlenden Werte bei continent2 beseitigt sind:

```
exp3 %>%
  filter(is.na(continent2))
```

```
## [1] Entity      Code      Year      e10      continent continent2
## <0 rows> (or 0-length row.names)
```

Gut.

Was ist die mittlere Lebenserwartung pro Kontinent im Jahr 2015 (laut den Daten)?

Für welche Jahre liegen Daten vor?

```
exp3 %>%
  distinct(Year) %>%
  pull()
```

```
## [1] 1950 1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010 2015
## [15] 2020 2025 2030 2035 2040 2045 2050 2055 2060 2065 2070 2075 2080 2085
## [29] 2090 2095 1750 1755 1760 1765 1770 1775 1780 1785 1790 1795 1800 1805
## [43] 1810 1815 1820 1825 1830 1835 1840 1845 1850 1855 1860 1865 1870 1875
## [57] 1880 1885 1890 1895 1900 1905 1910 1915 1920 1925 1930 1935 1940 1945
```

Möchte man die Daten nicht als Tabelle haben (braucht weniger Platz im Output), sondern aus der Tabelle eine Spalte *herausziehen*, kann man dafür den Befehl `pull()` verwenden.

Man sieht, dass die Daten offenbar nur in Fünf-Jahres-Schritten erhoben wurden. Wählen wir das Jahr 2015:

```
exp_2015 <- filter(exp3, Year == 2015)
```

Deskriptivstatistik

Berechnen wir die typischen Deskriptivstatistiken pro Kontinent:

```
favstats(e10 ~ continent2, data = exp_2015)
```

```
## continent2 min      Q1 median      Q3      max      mean      sd  n
## 1      Africa 43.30 55.8800 57.680 60.6300 71.13 58.29561 5.504426 57
## 2    Americas 59.20 65.7200 67.395 69.8175 73.58 67.47950 3.431801 40
## 3        Asia 57.81 62.4900 65.795 68.1125 74.73 65.75000 4.185182 50
## 4      Europe 61.12 66.4975 71.190 72.0950 73.97 69.27725 3.720629 40
## 5     Oceania 57.36 61.8900 65.160 68.4000 73.37 65.63615 4.852655 13
## 6        World 65.31 65.3100 65.310 65.3100 65.31 65.31000      NA  1
## missing
## 1          0
## 2          0
## 3          0
## 4          0
## 5          0
## 6          0
```

Welche Länder gehören eigentlich zu “Oceania”?

```
filter(exp_2015, continent == "Oceania") %>%  
  pull(Entity)
```

```
## [1] Australia      Fiji             French Polynesia Guam  
## [5] Kiribati        New Caledonia   New Zealand     Papua New Guinea  
## [9] Samoa          Solomon Islands Tonga            Vanuatu  
## 201 Levels: Afghanistan Albania Algeria Angola ... Zimbabwe
```

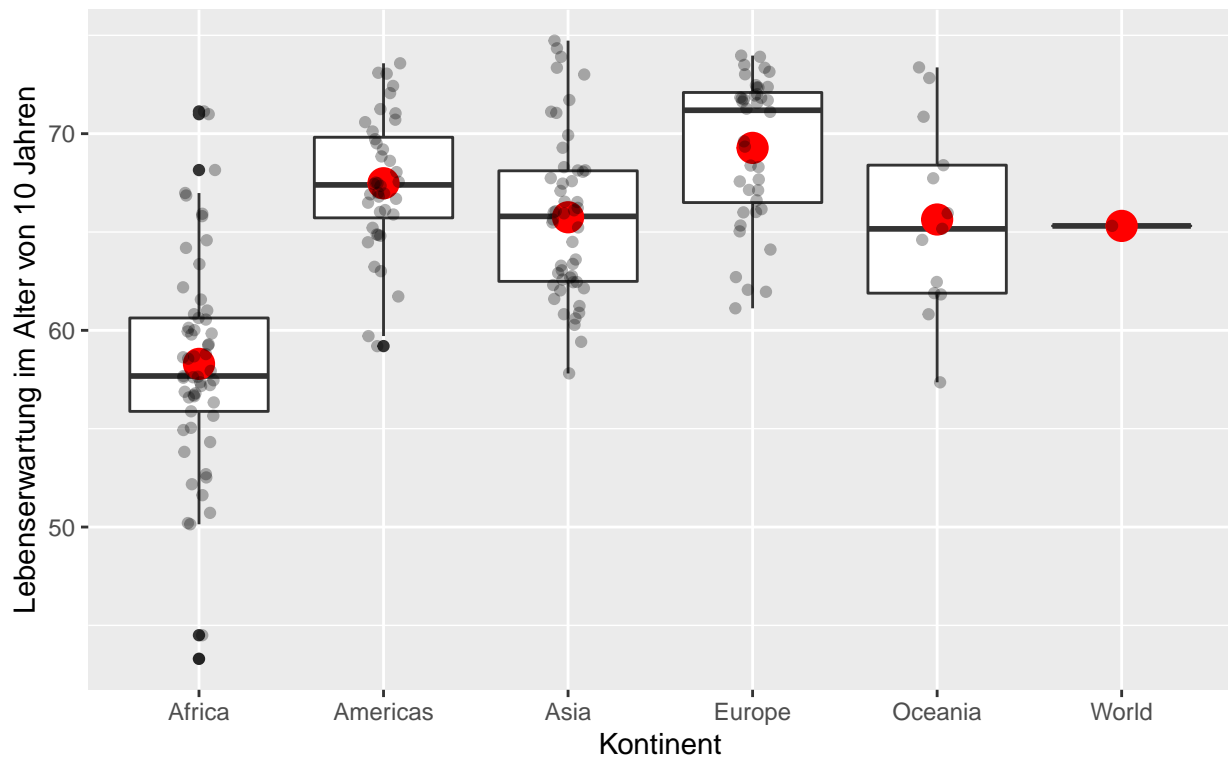
Vermutlich würde es Sinn machen, Nord- und Südamerika getrennt auszuweisen.

Visualisierung

```
gf_boxplot(e10 ~ continent2,  
  data = exp_2015) %>%  
  gf_point(stat = "summary",  
    color = "red",  
    size = 5) %>%  
  gf_jitter(width = .1,  
    alpha = .3) %>%  
  gf_labs(x= "Kontinent", y = "Lebenserwartung im Alter von 10 Jahren",  
    caption = "Daten von 'Our World in Data'",  
    title = "Lebenserwartung nach Kontinenten im Jahr 2015")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

Lebenserwartung nach Kontinenten im Jahr 2015



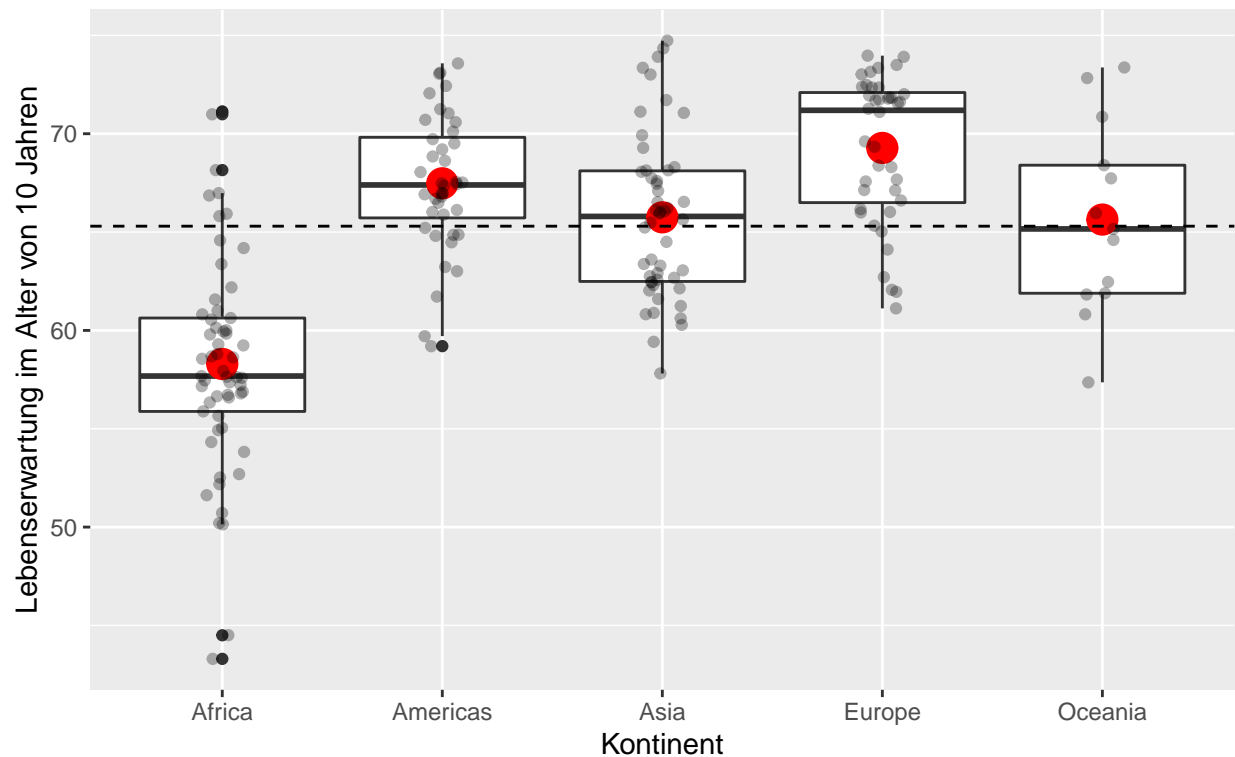
Daten von 'Our World in Data'

Da World kein Kontinent ist, nehmen wir es heraus, aber zeichnen eine horizontale Linie für den Wert der Welt insgesamt.

```
exp_2015 %>%
  filter(continent2 != "World") %>%
  gf_boxplot(e10 ~ continent2) %>%
  gf_point(stat = "summary",
           color = "red",
           size = 5) %>%
  gf_jitter(width = .1,
            alpha = .3) %>%
  gf_labs(x = "Kontinent", y = "Lebenserwartung im Alter von 10 Jahren",
          caption = "Daten von 'Our World in Data'. Die horizontale Linie gibt den weltweiten Wert wieder.",
          title = "Lebenserwartung nach Kontinenten im Jahr 2015.") %>%
  gf_hline(yintercept = ~65.3, linetype = "dashed")
```

```
## No summary function supplied, defaulting to `mean_se()`
```

Lebenserwartung nach Kontinenten im Jahr 2015.



Daten von 'Our World in Data'. Die horizontale Linie gibt den weltweiten Wert wieder.

Extremwerte

Welche drei Länder weisen die geringste/höchste Lebenserwartung im Jahr 2015 auf?

```
exp_2015 %>%
  arrange(e10) %>% # sortiert von klein nach groß bzgl. e10
  slice(1:3)
```

```
##      Entity Code Year  e10 continent continent2
## 1  Swaziland  SWZ  2015 43.30   Africa   Africa
## 2   Lesotho   LSO  2015 44.50   Africa   Africa
## 3 Cote d'Ivoire CIV  2015 50.14   Africa   Africa
```

slice() "schneidet" eine "Scheibe" an Zeilen heraus, hier 1 bis 3.

In gleicher Manier:

```
exp_2015 %>%
  arrange(-e10) %>%
  slice
```

```
##      Entity Code Year  e10 continent
## 1  Hong Kong   HKG  2015 74.73   Asia
## 2    Japan     JPN  2015 74.34   Asia
```


## 3	Italy	ITA	2015	73.97	Europe
## 4	Singapore	SGP	2015	73.91	Asia
## 5	Switzerland	CHE	2015	73.91	Europe
## 6	Chile	CHL	2015	73.58	Americas
## 7	Spain	ESP	2015	73.50	Europe
## 8	Australia	AUS	2015	73.37	Oceania
## 9	Iceland	ISL	2015	73.35	Europe
## 10	Israel	ISR	2015	73.35	Asia
## 11	France	FRA	2015	73.15	Europe
## 12	Martinique	MTQ	2015	73.10	Americas
## 13	Canada	CAN	2015	73.05	Americas
## 14	Sweden	SWE	2015	73.02	Europe
## 15	South Korea	KOR	2015	73.01	Asia
## 16	New Zealand	NZL	2015	72.83	Oceania
## 17	Luxembourg	LUX	2015	72.47	Europe
## 18	Guadeloupe	GLP	2015	72.43	Americas
## 19	Netherlands	NLD	2015	72.38	Europe
## 20	Austria	AUT	2015	72.35	Europe
## 21	Norway	NOR	2015	72.35	Europe
## 22	United States Virgin Islands	VIR	2015	72.06	Americas
## 23	Portugal	PRT	2015	72.01	Europe
## 24	Channel Islands	OWID_CIS	2015	71.94	<NA>
## 25	Greece	GRC	2015	71.84	Europe
## 26	Germany	DEU	2015	71.82	Europe
## 27	Ireland	IRL	2015	71.79	Europe
## 28	Belgium	BEL	2015	71.74	Europe
## 29	Macao	MAC	2015	71.71	Asia
## 30	Finland	FIN	2015	71.70	Europe
## 31	United Kingdom	GBR	2015	71.62	Europe
## 32	Malta	MLT	2015	71.56	Europe
## 33	Slovenia	SVN	2015	71.27	Europe
## 34	French Guiana	GUF	2015	71.25	Americas
## 35	Reunion	REU	2015	71.13	Africa
## 36	Cyprus	CYP	2015	71.12	Asia
## 37	Denmark	DNK	2015	71.11	Europe
## 38	Lebanon	LBN	2015	71.06	Asia
## 39	Costa Rica	CRI	2015	71.04	Americas
## 40	Mayotte	MYT	2015	70.99	Africa
## 41	Guam	GUM	2015	70.86	Oceania
## 42	Puerto Rico	PRI	2015	70.71	Americas
## 43	Cuba	CUB	2015	70.59	Americas
## 44	United States	USA	2015	70.11	Americas
## 45	Brunei	BRN	2015	69.92	Asia
## 46	Panama	PAN	2015	69.72	Americas
## 47	Albania	ALB	2015	69.61	Europe
## 48	Curacao	CUW	2015	69.51	Americas
## 49	Czech Republic	CZE	2015	69.34	Europe
## 50	Qatar	QAT	2015	69.28	Asia
## 51	Mexico	MEX	2015	69.20	Americas
## 52	Uruguay	URY	2015	68.84	Americas
## 53	Ecuador	ECU	2015	68.62	Americas
## 54	New Caledonia	NCL	2015	68.40	Oceania
## 55	Poland	POL	2015	68.38	Europe
## 56	Croatia	HRV	2015	68.30	Europe

## 57	Maldives	MDV	2015	68.30	Asia
## 58	Algeria	DZA	2015	68.15	Africa
## 59	Oman	OMN	2015	68.14	Asia
## 60	Vietnam	VNM	2015	68.13	Asia
## 61	United Arab Emirates	ARE	2015	68.06	Asia
## 62	Argentina	ARG	2015	68.05	Americas
## 63	Bahrain	BHR	2015	67.74	Asia
## 64	French Polynesia	PYF	2015	67.73	Oceania
## 65	Bosnia and Herzegovina	BIH	2015	67.67	Europe
## 66	Turkey	TUR	2015	67.59	Asia
## 67	Estonia	EST	2015	67.57	Europe
## 68	Peru	PER	2015	67.55	Americas
## 69	Nicaragua	NIC	2015	67.52	Americas
## 70	Antigua and Barbuda	ATG	2015	67.48	Americas
## 71	China	CHN	2015	67.46	Asia
## 72	Jamaica	JAM	2015	67.44	Americas
## 73	Aruba	ABW	2015	67.35	Americas
## 74	Slovakia	SVK	2015	67.14	Europe
## 75	Montenegro	MNE	2015	67.12	Europe
## 76	Iran	IRN	2015	67.09	Asia
## 77	Brazil	BRA	2015	66.99	Americas
## 78	Tunisia	TUN	2015	66.98	Africa
## 79	Barbados	BRB	2015	66.96	Americas
## 80	Bahamas	BHS	2015	66.92	Americas
## 81	Morocco	MAR	2015	66.86	Africa
## 82	Honduras	HND	2015	66.80	Americas
## 83	Saint Lucia	LCA	2015	66.69	Americas
## 84	Macedonia	MKD	2015	66.61	Europe
## 85	Armenia	ARM	2015	66.54	Asia
## 86	Georgia	GEO	2015	66.52	Asia
## 87	Colombia	COL	2015	66.49	Americas
## 88	Sri Lanka	LKA	2015	66.23	Asia
## 89	Serbia	SRB	2015	66.16	Europe
## 90	Dominican Republic	DOM	2015	66.12	Americas
## 91	Thailand	THA	2015	66.10	Asia
## 92	Saudi Arabia	SAU	2015	66.06	Asia
## 93	Romania	ROU	2015	66.02	Europe
## 94	Venezuela	VEN	2015	66.02	Americas
## 95	Hungary	HUN	2015	66.00	Europe
## 96	Malaysia	MYS	2015	66.00	Asia
## 97	Samoa	WSM	2015	65.96	Oceania
## 98	Jordan	JOR	2015	65.95	Asia
## 99	Mauritius	MUS	2015	65.93	Africa
## 100	Paraguay	PRY	2015	65.89	Americas
## 101	Cape Verde	CPV	2015	65.81	Africa
## 102	Kuwait	KWT	2015	65.64	Asia
## 103	Bangladesh	BGD	2015	65.48	Asia
## 104	Bulgaria	BGR	2015	65.33	Europe
## 105	World	OWID_WRL	2015	65.31	<NA>
## 106	Palestine	PSE	2015	65.23	Asia
## 107	El Salvador	SLV	2015	65.21	Americas
## 108	Tonga	TON	2015	65.16	Oceania
## 109	Latvia	LVA	2015	65.03	Europe
## 110	Saint Vincent and the Grenadines	VCT	2015	64.87	Americas

## 111	Grenada	GRD	2015	64.86	Americas
## 112	Guatemala	GTM	2015	64.81	Americas
## 113	Vanuatu	VUT	2015	64.60	Oceania
## 114	Seychelles	SYC	2015	64.58	Africa
## 115	Azerbaijan	AZE	2015	64.50	Asia
## 116	Bolivia	BOL	2015	64.48	Americas
## 117	Libya	LBY	2015	64.19	Africa
## 118	Lithuania	LTU	2015	64.11	Europe
## 119	Tajikistan	TJK	2015	63.60	Asia
## 120	Bhutan	BTN	2015	63.38	Asia
## 121	Egypt	EGY	2015	63.37	Africa
## 122	Nepal	NPL	2015	63.29	Asia
## 123	Suriname	SUR	2015	63.23	Americas
## 124	North Korea	PRK	2015	63.05	Asia
## 125	Trinidad and Tobago	TTO	2015	63.01	Americas
## 126	Timor	TLS	2015	62.91	<NA>
## 127	Pakistan	PAK	2015	62.76	Asia
## 128	Moldova	MDA	2015	62.71	Europe
## 129	Uzbekistan	UZB	2015	62.68	Asia
## 130	Iraq	IRQ	2015	62.58	Asia
## 131	Kyrgyzstan	KGZ	2015	62.46	Asia
## 132	Micronesia (country)	FSM	2015	62.46	<NA>
## 133	India	IND	2015	62.45	Asia
## 134	Mongolia	MNG	2015	62.31	Asia
## 135	Western Sahara	ESH	2015	62.19	Africa
## 136	Cambodia	KHM	2015	62.14	Asia
## 137	Belarus	BLR	2015	62.06	Europe
## 138	Syria	SYR	2015	62.03	Asia
## 139	Ukraine	UKR	2015	61.96	Europe
## 140	Fiji	FJI	2015	61.89	Oceania
## 141	Solomon Islands	SLB	2015	61.83	Oceania
## 142	Belize	BLZ	2015	61.72	Americas
## 143	Indonesia	IDN	2015	61.60	Asia
## 144	Sao Tome and Principe	STP	2015	61.57	Africa
## 145	Laos	LAO	2015	61.24	Asia
## 146	Russia	RUS	2015	61.12	Europe
## 147	Senegal	SEN	2015	61.01	Africa
## 148	Kazakhstan	KAZ	2015	60.89	Asia
## 149	Kiribati	KIR	2015	60.82	Oceania
## 150	Philippines	PHL	2015	60.82	Asia
## 151	Rwanda	RWA	2015	60.82	Africa
## 152	Malawi	MWI	2015	60.63	Africa
## 153	Myanmar	MMR	2015	60.61	Asia
## 154	Ethiopia	ETH	2015	60.55	Africa
## 155	Turkmenistan	TKM	2015	60.28	Asia
## 156	Gabon	GAB	2015	60.13	Africa
## 157	Madagascar	MDG	2015	60.01	Africa
## 158	Sudan	SDN	2015	59.95	Africa
## 159	Mauritania	MRT	2015	59.84	Africa
## 160	Tanzania	TZA	2015	59.79	Africa
## 161	Guyana	GUY	2015	59.71	Americas
## 162	Yemen	YEM	2015	59.42	Asia
## 163	Comoros	COM	2015	59.29	Africa
## 164	Niger	NER	2015	59.24	Africa

## 165	Haiti	HTI	2015	59.20	Americas
## 166	Congo	COG	2015	58.79	Africa
## 167	Kenya	KEN	2015	58.68	Africa
## 168	Djibouti	DJI	2015	58.63	Africa
## 169	Eritrea	ERI	2015	58.56	Africa
## 170	Namibia	NAM	2015	57.94	Africa
## 171	Afghanistan	AFG	2015	57.81	Asia
## 172	Democratic Republic of Congo	COD	2015	57.68	Africa
## 173	Benin	BEN	2015	57.64	Africa
## 174	Liberia	LBR	2015	57.62	Africa
## 175	Zambia	ZMB	2015	57.59	Africa
## 176	Ghana	GHA	2015	57.46	Africa
## 177	Papua New Guinea	PNG	2015	57.36	Oceania
## 178	Uganda	UGA	2015	57.36	Africa
## 179	Mali	MLI	2015	57.22	Africa
## 180	Botswana	BWA	2015	57.17	Africa
## 181	Burkina Faso	BFA	2015	56.88	Africa
## 182	Gambia	GMB	2015	56.80	Africa
## 183	Zimbabwe	ZWE	2015	56.72	Africa
## 184	Togo	TGO	2015	56.65	Africa
## 185	Guinea	GIN	2015	56.59	Africa
## 186	Burundi	BDI	2015	56.34	Africa
## 187	Guinea-Bissau	GNB	2015	55.88	Africa
## 188	Equatorial Guinea	GNQ	2015	55.66	Africa
## 189	Somalia	SOM	2015	55.04	Africa
## 190	South Sudan	SSD	2015	54.93	Africa
## 191	Cameroon	CMR	2015	54.32	Africa
## 192	Angola	AGO	2015	53.82	Africa
## 193	Central African Republic	CAF	2015	52.69	Africa
## 194	Chad	TCD	2015	52.52	Africa
## 195	Mozambique	MOZ	2015	52.18	Africa
## 196	Nigeria	NGA	2015	51.62	Africa
## 197	South Africa	ZAF	2015	50.72	Africa
## 198	Sierra Leone	SLE	2015	50.20	Africa
## 199	Cote d'Ivoire	CIV	2015	50.14	Africa
## 200	Lesotho	LSO	2015	44.50	Africa
## 201	Swaziland	SWZ	2015	43.30	Africa
##	continent2				
## 1	Asia				
## 2	Asia				
## 3	Europe				
## 4	Asia				
## 5	Europe				
## 6	Americas				
## 7	Europe				
## 8	Oceania				
## 9	Europe				
## 10	Asia				
## 11	Europe				
## 12	Americas				
## 13	Americas				
## 14	Europe				
## 15	Asia				
## 16	Oceania				

17 Europe
18 Americas
19 Europe
20 Europe
21 Europe
22 Americas
23 Europe
24 Europe
25 Europe
26 Europe
27 Europe
28 Europe
29 Asia
30 Europe
31 Europe
32 Europe
33 Europe
34 Americas
35 Africa
36 Asia
37 Europe
38 Asia
39 Americas
40 Africa
41 Oceania
42 Americas
43 Americas
44 Americas
45 Asia
46 Americas
47 Europe
48 Americas
49 Europe
50 Asia
51 Americas
52 Americas
53 Americas
54 Oceania
55 Europe
56 Europe
57 Asia
58 Africa
59 Asia
60 Asia
61 Asia
62 Americas
63 Asia
64 Oceania
65 Europe
66 Asia
67 Europe
68 Americas
69 Americas
70 Americas

71 Asia
72 Americas
73 Americas
74 Europe
75 Europe
76 Asia
77 Americas
78 Africa
79 Americas
80 Americas
81 Africa
82 Americas
83 Americas
84 Europe
85 Asia
86 Asia
87 Americas
88 Asia
89 Europe
90 Americas
91 Asia
92 Asia
93 Europe
94 Americas
95 Europe
96 Asia
97 Oceania
98 Asia
99 Africa
100 Americas
101 Africa
102 Asia
103 Asia
104 Europe
105 World
106 Asia
107 Americas
108 Oceania
109 Europe
110 Americas
111 Americas
112 Americas
113 Oceania
114 Africa
115 Asia
116 Americas
117 Africa
118 Europe
119 Asia
120 Asia
121 Africa
122 Asia
123 Americas
124 Asia

125 Americas
126 Asia
127 Asia
128 Europe
129 Asia
130 Asia
131 Asia
132 Oceania
133 Asia
134 Asia
135 Africa
136 Asia
137 Europe
138 Asia
139 Europe
140 Oceania
141 Oceania
142 Americas
143 Asia
144 Africa
145 Asia
146 Europe
147 Africa
148 Asia
149 Oceania
150 Asia
151 Africa
152 Africa
153 Asia
154 Africa
155 Asia
156 Africa
157 Africa
158 Africa
159 Africa
160 Africa
161 Americas
162 Asia
163 Africa
164 Africa
165 Americas
166 Africa
167 Africa
168 Africa
169 Africa
170 Africa
171 Asia
172 Africa
173 Africa
174 Africa
175 Africa
176 Africa
177 Oceania
178 Africa

```
## 179 Africa
## 180 Africa
## 181 Africa
## 182 Africa
## 183 Africa
## 184 Africa
## 185 Africa
## 186 Africa
## 187 Africa
## 188 Africa
## 189 Africa
## 190 Africa
## 191 Africa
## 192 Africa
## 193 Africa
## 194 Africa
## 195 Africa
## 196 Africa
## 197 Africa
## 198 Africa
## 199 Africa
## 200 Africa
## 201 Africa
```

Das Minuszeichen dreht die Sortierreihenfolge um, d.h. von groß zu klein.

Gibt es einen Zusammenhang von Lebenserwartung und dem Jahr?

Deskriptivstatistik

```
cor(e10 ~ Year, data = exp3)
```

```
## [1] NA
```

Oh, es gibt fehlende Werte im Datensatz, daher streckt der Befehl `cor()` alle Viere von sich. Löschen wir mal alle Zeilen mit fehlenden Werten und hoffen, das wir nicht viele Daten verlieren:

```
exp_ohne_na <- exp3 %>%
  na.omit()
```

Hm, ca. 500 Zeilen. Vielleicht doch lieber so:

```
cor(e10 ~ Year, data = exp3 %>% na.omit())
```

```
## [1] 0.7531835
```

Inferenzstatistik

Inferenzstatistik macht hier wenig Sinn, da es unklar klar ist, auf welche Grundgesamtheit verallgemeinert werden soll. Alle Länder der Welt sind ja schon enthalten. Zu argumentieren, wir verallgemeinern auf alle Zeiten (Jahre) ist gewagt, denn unsere Stichprobe an Jahren ist sicher alles andere als eine Zufallsstichprobe aus der Menge aller Jahre der Welt ... Verzichten wir also auf eine Inferenzstatistik.

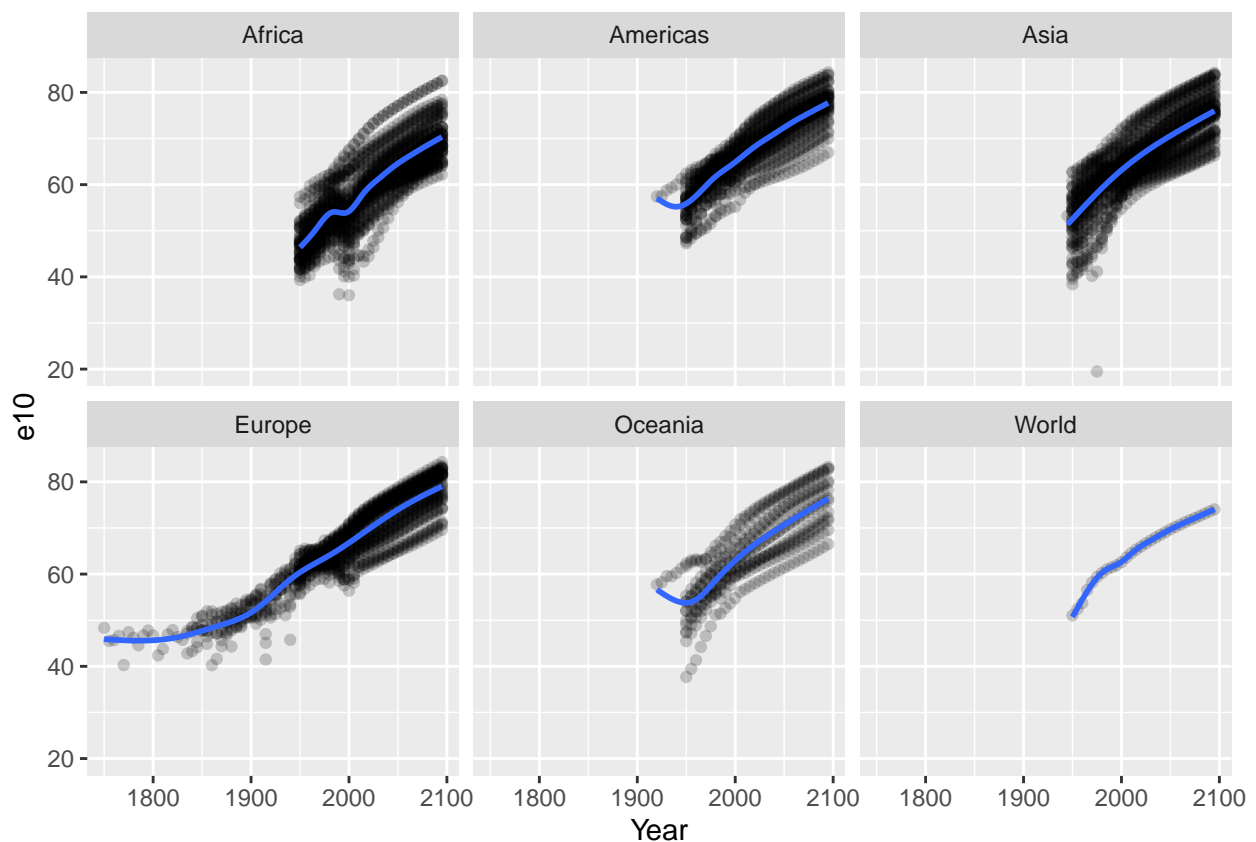
Visualisierung

```
gf_point(e10 ~ Year, data = exp3, alpha = .2) %>%  
  gf_smooth() %>%  
  gf_facet_wrap(~ continent2)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 419 rows containing non-finite values (stat_smooth).
```

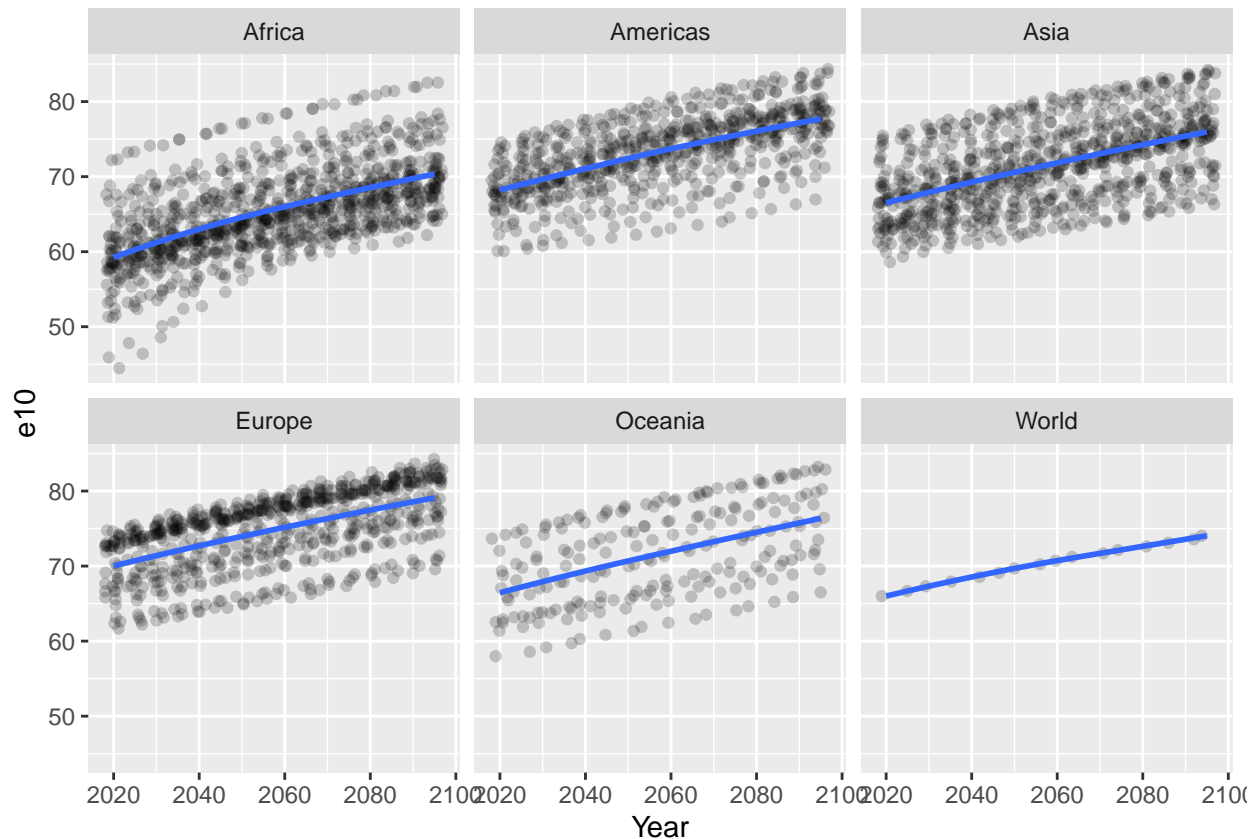
```
## Warning: Removed 419 rows containing missing values (geom_point).
```



`gf_facet_wrap()` macht ein “Teil-Bildchen” (Facette) pro Wert von Kontinent. Allerdings liegen die Daten für die meisten Länder erst ab ca. 1950 vor. Und die Prognosen in die Zukunft sehen wir mal kritisch. Vorhersagen sind bekanntlich schwierig. Vor allem, wenn sie die Zukunft betreffen, heißt es ... `gf_smooth()` legt eine Kurve in “die Mitte” des Punkteschwarm: Für jeden X-Werte wird der mittlere Y-Wert berechnet (nahe X-Werte fließen auch noch etwas ein), und dann werden die Punkte mit einer Linie verbunden.

```
exp3 %>%  
  filter(Year > 2016 & Year > 1949) %>%  
  gf_jitter(e10 ~ Year, alpha = .2) %>%  
  gf_smooth() %>%  
  gf_facet_wrap(~ continent2)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Wir sehen durchweg einen Anstieg der Lebenserwartung - sehr erfreulich! Freilich fußt der Anstieg auf unterschiedlichen Sockeln, also Ausgangsniveaus. Ist die Höhe des Anstiegs unterschiedlich je Kontinent? Für diese Frage modellieren wir die Lebenserwartung als Funktion des Kontinents und des Jahres.

Bleiben wir beim realistischen Teil der Daten:

```
exp_1950_2015 <- exp3 %>%
  filter(Year > 2016 & Year < 1949)
```

Modellierung

Berechnen wir dazu einige Modelle und vergleichen diese dann:

1. Das Nullmodell: Kein Prädiktor
2. Nur Year als Prädiktor
3. Year und continent2, aber ohne Interaktionseffekt
4. Year*continent2, also mit Interaktionseffekt

```
lm0 <- lm(e10 ~ 1, data = exp_1950_2015)
lm1 <- lm(e10 ~ Year, data = exp_1950_2015)
lm2 <- lm(e10 ~ Year + continent2, data = exp_1950_2015)
lm3 <- lm(e10 ~ Year*continent2, data = exp_1950_2015)
```

Betrachten wir die R^2 -Werte jedes Modells:

```
rsquared(lm0) %>% str()
```

```
## num 0
```

```
r2s_vector <- c(rsquared(lm0), rsquared(lm1), rsquared(lm2), rsquared(lm3))
r2s_vector %>% str()
```

```
## num [1:4] 0 0.219 0.531 0.532
```

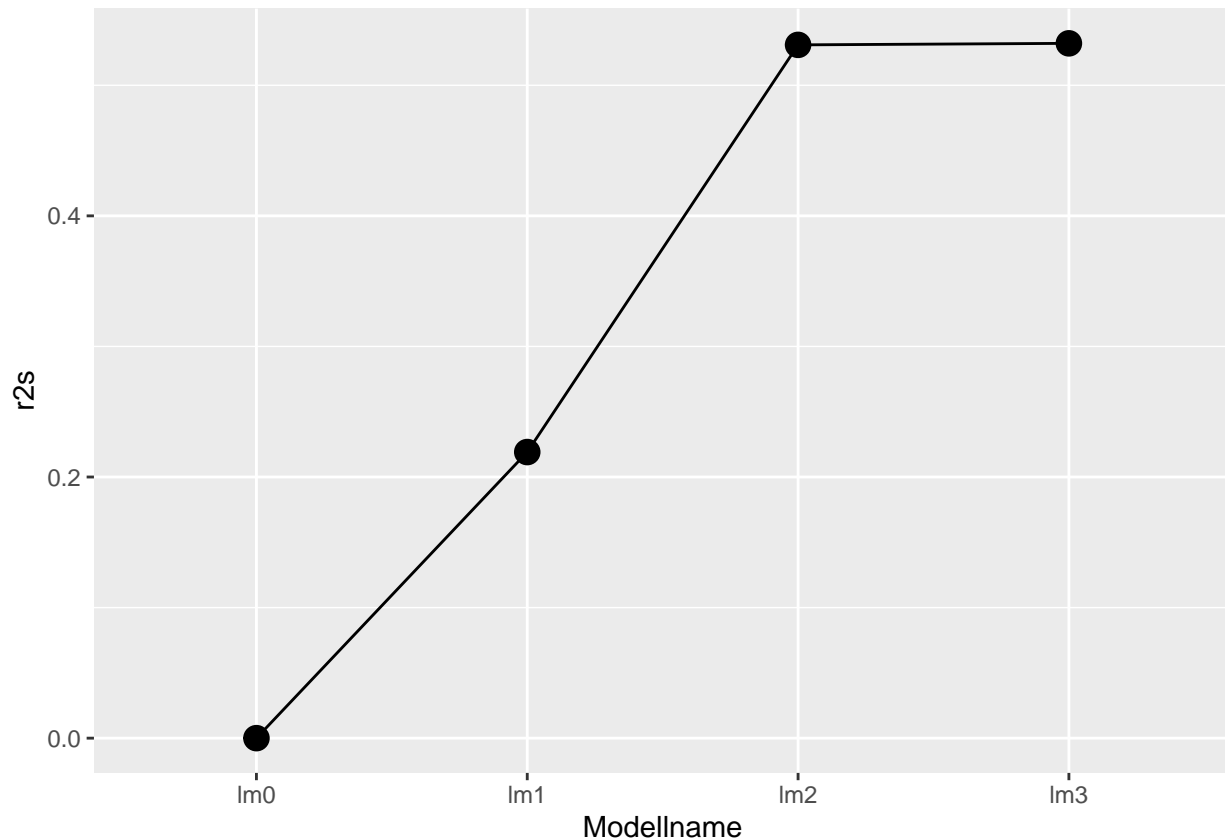
Um die Werte in einer kleiner Grafik zu zeigen, erstellen wir zuerst eine Tabelle (`tibble()`), weil `gf_XXX()` nur Spaß an Tabellen hat.

```
r2s <- tibble(Modellname = c("lm0", "lm1", "lm2", "lm3"),
              r2s = r2s_vector)
r2s
```

```
## # A tibble: 4 x 2
##   Modellname    r2s
##   <chr>        <dbl>
## 1 lm0          0
## 2 lm1        0.219
## 3 lm2        0.531
## 4 lm3        0.532
```

Und jetzt das Diagramm:

```
gf_point(r2s ~ Modellname, data = r2s, size = 4) %>%
  gf_line(group = ~1) # Es gibt nur eine Gruppe, also alle Punkte sollen verbunden werden
```



Wie man sieht, ist der Zuwachs an erklärter Varianz von `lm2` auf `lm3` nicht groß. Der Interaktionseffekt scheint also nicht stark zu sein. Besser wir verzichten auf ihn und resümieren, dass die Daten eher die Hypothese stützen, dass der Zuwachs in Lebenserwartung gleich oder ähnlich groß ist zwischen den Kontinenten.

Wir können die Güte der Modelle bzw. die *Unterschiede* der Güte auf Signifikanz testen. Genau genommen werden die Likelihoods der Modelle verglichen und anhand der Freiheitsgrade normiert. Aber lassen wir die technischen Details an dieser Stelle. Diesen Modellvergleich können anhand der Funktion `anova()` durchführen, der uns die Quadratsummen zurückliefert und einen p-Wert (letzte Spalte).

```
anova(lm0, lm1, lm2, lm3)
```

```
## Analysis of Variance Table
##
## Model 1: e10 ~ 1
## Model 2: e10 ~ Year
## Model 3: e10 ~ Year + continent2
## Model 4: e10 ~ Year * continent2
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    3215 133019
## 2    3214 103875  1    29145 1500.3578 <2e-16 ***
## 3    3209  62394  5    41481  427.0882 <2e-16 ***
## 4    3204  62238  5        156    1.6039 0.1555
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Wir sehen, dass das letzte Modell (Model 4) nicht signifikant besser ist als das vorherige Modell. Das bestärkt unseren Schluss, die H_0 der Gleichheit der Modellgüte der beiden Modelle, nicht zu verwerfen.

Geovisualisierung

Weltkarte zeichnen

Analysiert man Länder (oder allgemeiner: Gegenden) der Erde, so bietet sich eine Geo-Visualisierung - wie eine Weltkarte - an. Das lässt sich recht einfach mit R bewerkstelligen.

Im Paket `rnaturalearth` sind die Weltkarten gespeichert. Ziehen wir uns die Länder der “natural earth” (ne) zunächst heraus:

```
world <- ne_countries(scale = "medium", returnclass = "sf")
```

Die Tabelle ist sehr umfangreich; viele interessante Daten sind vorhanden. Die Geo-Daten verstecken sich in der Spalte `geometry`. Die technischen Details interessieren uns hier nicht.

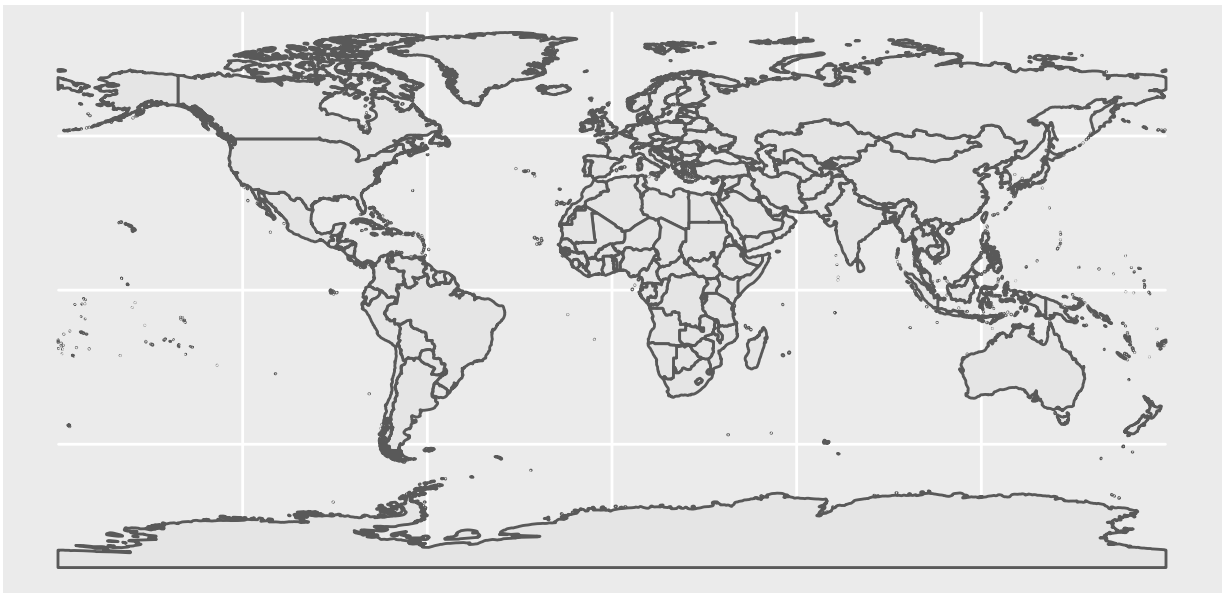
```
str(world)
```

```
## Classes 'sf' and 'data.frame':  241 obs. of  64 variables:
## $ scalerank : int  3 1 1 1 1 3 3 1 1 1 ...
## $ featurecla: chr  "Admin-0 country" "Admin-0 country" "Admin-0 country" "Admin-0 country" ...
## $ labelrank : num  5 3 3 6 6 6 6 4 2 6 ...
## $ sovereignt: chr  "Netherlands" "Afghanistan" "Angola" "United Kingdom" ...
## $ sov_a3     : chr  "NL1" "AFG" "AGO" "GB1" ...
## $ adm0_dif   : num  1 0 0 1 0 1 0 0 0 0 ...
## $ level      : num  2 2 2 2 2 2 2 2 2 2 ...
## $ type       : chr  "Country" "Sovereign country" "Sovereign country" "Dependency" ...
## $ admin      : chr  "Aruba" "Afghanistan" "Angola" "Anguilla" ...
## $ adm0_a3    : chr  "ABW" "AFG" "AGO" "AIA" ...
## $ geou_dif   : num  0 0 0 0 0 0 0 0 0 0 ...
## $ geounit    : chr  "Aruba" "Afghanistan" "Angola" "Anguilla" ...
## $ gu_a3      : chr  "ABW" "AFG" "AGO" "AIA" ...
## $ su_dif     : num  0 0 0 0 0 0 0 0 0 0 ...
## $ subunit    : chr  "Aruba" "Afghanistan" "Angola" "Anguilla" ...
## $ su_a3      : chr  "ABW" "AFG" "AGO" "AIA" ...
## $ brk_diff   : num  0 0 0 0 0 0 0 0 0 0 ...
## $ name       : chr  "Aruba" "Afghanistan" "Angola" "Anguilla" ...
## $ name_long  : chr  "Aruba" "Afghanistan" "Angola" "Anguilla" ...
## $ brk_a3     : chr  "ABW" "AFG" "AGO" "AIA" ...
## $ brk_name   : chr  "Aruba" "Afghanistan" "Angola" "Anguilla" ...
## $ brk_group  : chr  NA NA NA NA ...
## $ abbrev     : chr  "Aruba" "Afg." "Ang." "Ang." ...
## $ postal     : chr  "AW" "AF" "AO" "AI" ...
## $ formal_en  : chr  "Aruba" "Islamic State of Afghanistan" "People's Republic of Angola" NA ...
## $ formal_fr  : chr  NA NA NA NA ...
## $ note_adm0  : chr  "Neth." NA NA "U.K." ...
## $ note_brk   : chr  NA NA NA NA ...
## $ name_sort  : chr  "Aruba" "Afghanistan" "Angola" "Anguilla" ...
## $ name_alt   : chr  NA NA NA NA ...
## $ mapcolor7  : num  4 5 3 6 1 4 1 2 3 3 ...
## $ mapcolor8  : num  2 6 2 6 4 1 4 1 1 1 ...
## $ mapcolor9  : num  2 8 6 6 1 4 1 3 3 2 ...
## $ mapcolor13: num  9 7 1 3 6 6 8 3 13 10 ...
## $ pop_est    : num  103065 28400000 12799293 14436 3639453 ...
```

```
## $ gdp_md_est: num 2258 22270 110300 109 21810 ...
## $ pop_year : num NA NA NA NA NA NA NA NA NA NA ...
## $ lastcensus: num 2010 1979 1970 NA 2001 ...
## $ gdp_year : num NA NA NA NA NA NA NA NA NA NA ...
## $ economy : chr "6. Developing region" "7. Least developed region" "7. Least developed region" "6. Developing region"
## $ income_grp: chr "2. High income: nonOECD" "5. Low income" "3. Upper middle income" "3. Upper middle income"
## $ wikipedia : num NA NA NA NA NA NA NA NA NA NA ...
## $ fips_10 : chr NA NA NA NA ...
## $ iso_a2 : chr "AW" "AF" "AO" "AI" ...
## $ iso_a3 : chr "ABW" "AFG" "AGO" "AIA" ...
## $ iso_n3 : chr "533" "004" "024" "660" ...
## $ un_a3 : chr "533" "004" "024" "660" ...
## $ wb_a2 : chr "AW" "AF" "AO" NA ...
## $ wb_a3 : chr "ABW" "AFG" "AGO" NA ...
## $ woe_id : num NA NA NA NA NA NA NA NA NA NA ...
## $ adm0_a3_is: chr "ABW" "AFG" "AGO" "AIA" ...
## $ adm0_a3_us: chr "ABW" "AFG" "AGO" "AIA" ...
## $ adm0_a3_un: num NA NA NA NA NA NA NA NA NA NA ...
## $ adm0_a3_wb: num NA NA NA NA NA NA NA NA NA NA ...
## $ continent : chr "North America" "Asia" "Africa" "North America" ...
## $ region_un : chr "Americas" "Asia" "Africa" "Americas" ...
## $ subregion : chr "Caribbean" "Southern Asia" "Middle Africa" "Caribbean" ...
## $ region_wb : chr "Latin America & Caribbean" "South Asia" "Sub-Saharan Africa" "Latin America & Caribbean"
## $ name_len : num 5 11 6 8 7 5 7 20 9 7 ...
## $ long_len : num 5 11 6 8 7 13 7 20 9 7 ...
## $ abbrev_len: num 5 4 4 4 4 5 4 6 4 4 ...
## $ tiny : num 4 NA NA NA NA 5 5 NA NA NA ...
## $ homepart : num NA 1 1 NA 1 NA 1 1 1 1 ...
## $ geometry :sfc_MULTIPOLYGON of length 241; first list element: List of 1
## ..$ :List of 1
## .. ..$ : num [1:10, 1:2] -69.9 -69.9 -69.9 -70 -70.1 ...
## ..- attr(*, "class")= chr "XY" "MULTIPOLYGON" "sfg"
## - attr(*, "sf_column")= chr "geometry"
## - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",...: NA NA NA NA NA NA NA NA NA NA ...
## ..- attr(*, "names")= chr "scalerank" "featurecla" "labelrank" "sovereignty" ...
```

Wir haben den Befehl angewiesen, dass wir keine super genauen Grenzen benötigen, sondern dass uns eine mittlere Genauigkeit ausreicht. Das Datenformat nennt sich `sf` (simple feature), was zur Zeit ein Standardformat ist, um geografische Daten (GPS-Koordinaten für Grenzen und Punkte etc.) zu speichern. Aber wie gesagt, die technischen Details sind hier nicht interessant für uns. Mit dem Befehl `gf_sf` lassen sich dann Geo-Daten plotten:

```
gf_sf(data = world)
```



Karte einfärben (Choropleth)

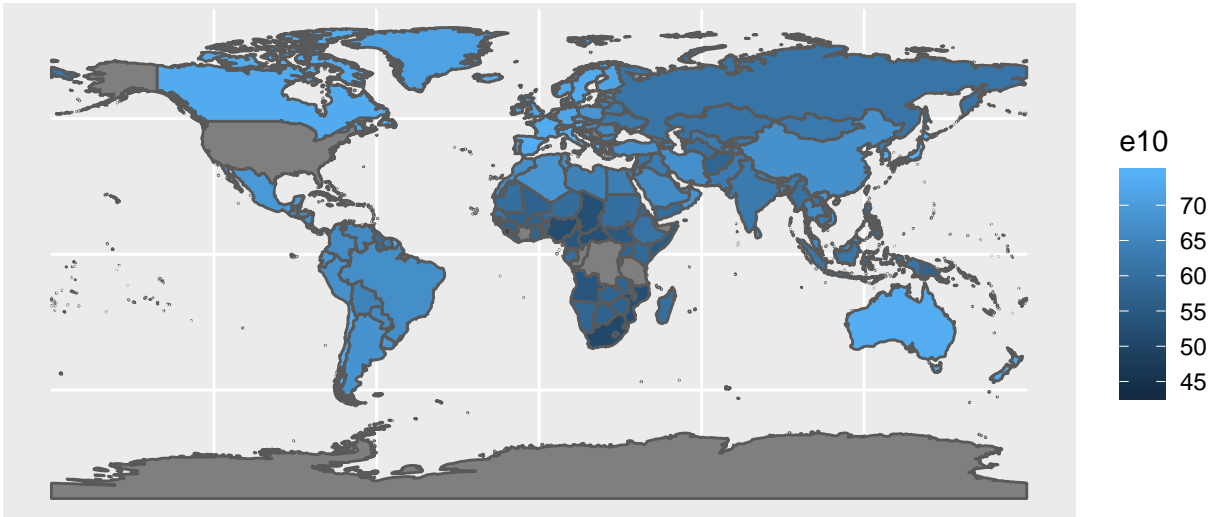
Damit wir die Länder entsprechend ihrer Lebenserwartung einfärben können, müssen wir die Tabelle mit den Geo-Daten und die Tabelle mit den Lebensdaten zusammenführen. Das besorgt der Befehl `full_join()`. Damit der Befehl weiß, welche Zeilen zusammen gehören, erklären wir ihm “Füge gleiche Länder zusammen, dazu schaue in der Spalte `Entity` bzw. in der Spalte `sovereignty` (Tabelle `world`)”.

```
exp_2015_joined <- world %>%  
  left_join(exp_2015, by = c("sovereignty" = "Entity"))
```

```
## Warning: Column `sovereignty`/`Entity` joining character vector and factor,  
## coercing into character vector
```

Das eigentliche Plotten ist schnell erledigt:

```
gf_sf(fill = ~e10, data = exp_2015_joined)
```

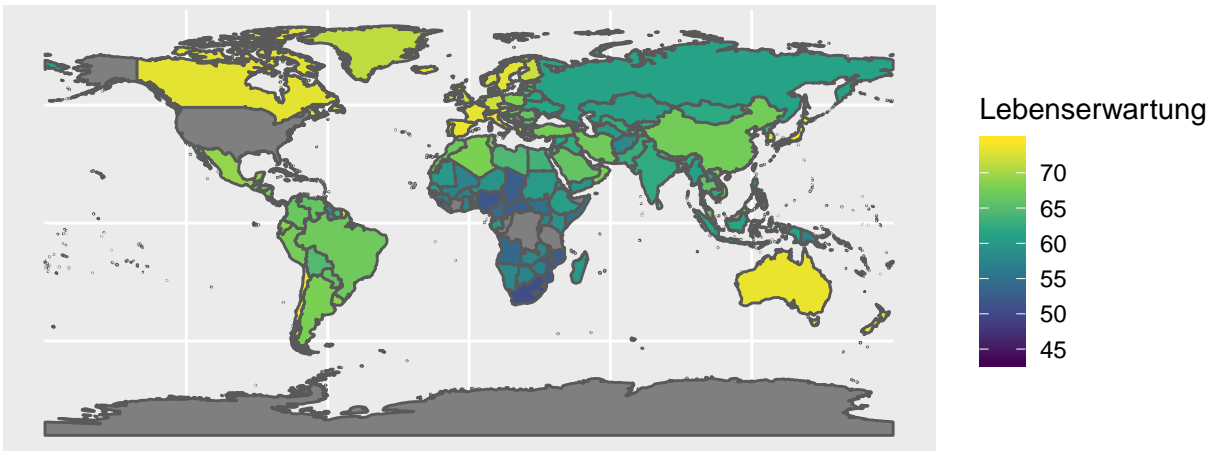


Praktischerweise findet `gg_sf()` die Geo-Daten selbständig in der Tabelle, wir müssen nicht extra erklären, welche Spalte gemeint ist. Sehr komfortabel.

Das Farbschema könnte noch schöner sein:

```
gg_sf(fill = ~e10, data = exp_2015_joined) %>%  
  gf_refine(scale_fill_viridis_c()) %>%  
  gf_labs(fill = "Lebenserwartung",  
          title = "Lebenserwartung im Alter von 10 Jahren")
```


Lebenserwartung im Alter von 10 Jahren



Viel besser. Mit `gf_refine()` ‘verfeinern‘ wir das Diagramm. In diesem Fall besteht das Verfeinern im Ändern des Farbschemas (wie nehmen *Viridis*), um die Füllfarbe zu ändern. Da es sich um eine kontinuierliche Variable handelt (d.h. metrisch), soll ein kontinuierliches Farbschema (mit fließenden, weichen Übergängen) verwendet werden ("`_c`" wie continuous).

Halt! Für die USA gibt es keine Werte! Kann das sein? Das ist ein gutes Beispiel dafür, dass eine Datenanalyse ein iterativer Prozess ist, d.h. einzelne Schritte - wie Datenvorverarbeitung - müssen immer wieder ausgeführt werden.

Auf der Suche nach den USA

Versuchen wir, die USA in den Daten zu finden:

```
exp3 %>%  
  filter(str_detect(Entity, "USA|United States"))
```

##	Entity	Code	Year	e10	continent	continent2
## 1	United States	USA	1950	61.28	Americas	Americas
## 2	United States	USA	1955	62.06	Americas	Americas
## 3	United States	USA	1960	62.32	Americas	Americas
## 4	United States	USA	1965	62.34	Americas	Americas
## 5	United States	USA	1970	62.94	Americas	Americas
## 6	United States	USA	1975	64.52	Americas	Americas
## 7	United States	USA	1980	65.43	Americas	Americas
## 8	United States	USA	1985	65.86	Americas	Americas

## 9	United States	USA	1990	66.49	Americas	Americas	
## 10	United States	USA	1995	67.16	Americas	Americas	
## 11	United States	USA	2000	67.83	Americas	Americas	
## 12	United States	USA	2005	68.79	Americas	Americas	
## 13	United States	USA	2010	69.47	Americas	Americas	
## 14	United States	USA	2015	70.11	Americas	Americas	
## 15	United States	USA	2020	70.80	Americas	Americas	
## 16	United States	USA	2025	71.50	Americas	Americas	
## 17	United States	USA	2030	72.19	Americas	Americas	
## 18	United States	USA	2035	72.94	Americas	Americas	
## 19	United States	USA	2040	73.65	Americas	Americas	
## 20	United States	USA	2045	74.30	Americas	Americas	
## 21	United States	USA	2050	74.88	Americas	Americas	
## 22	United States	USA	2055	75.45	Americas	Americas	
## 23	United States	USA	2060	76.00	Americas	Americas	
## 24	United States	USA	2065	76.51	Americas	Americas	
## 25	United States	USA	2070	77.04	Americas	Americas	
## 26	United States	USA	2075	77.55	Americas	Americas	
## 27	United States	USA	2080	78.06	Americas	Americas	
## 28	United States	USA	2085	78.54	Americas	Americas	
## 29	United States	USA	2090	79.03	Americas	Americas	
## 30	United States	USA	2095	79.52	Americas	Americas	
## 31	United States	Virgin Islands	VIR	1950	54.52	Americas	Americas
## 32	United States	Virgin Islands	VIR	1955	56.97	Americas	Americas
## 33	United States	Virgin Islands	VIR	1960	58.30	Americas	Americas
## 34	United States	Virgin Islands	VIR	1965	59.86	Americas	Americas
## 35	United States	Virgin Islands	VIR	1970	61.43	Americas	Americas
## 36	United States	Virgin Islands	VIR	1975	62.95	Americas	Americas
## 37	United States	Virgin Islands	VIR	1980	64.12	Americas	Americas
## 38	United States	Virgin Islands	VIR	1985	65.56	Americas	Americas
## 39	United States	Virgin Islands	VIR	1990	66.88	Americas	Americas
## 40	United States	Virgin Islands	VIR	1995	68.13	Americas	Americas
## 41	United States	Virgin Islands	VIR	2000	69.19	Americas	Americas
## 42	United States	Virgin Islands	VIR	2005	69.90	Americas	Americas
## 43	United States	Virgin Islands	VIR	2010	70.98	Americas	Americas
## 44	United States	Virgin Islands	VIR	2015	72.06	Americas	Americas
## 45	United States	Virgin Islands	VIR	2020	73.11	Americas	Americas
## 46	United States	Virgin Islands	VIR	2025	74.05	Americas	Americas
## 47	United States	Virgin Islands	VIR	2030	74.86	Americas	Americas
## 48	United States	Virgin Islands	VIR	2035	75.57	Americas	Americas
## 49	United States	Virgin Islands	VIR	2040	76.24	Americas	Americas
## 50	United States	Virgin Islands	VIR	2045	76.89	Americas	Americas
## 51	United States	Virgin Islands	VIR	2050	77.49	Americas	Americas
## 52	United States	Virgin Islands	VIR	2055	78.09	Americas	Americas
## 53	United States	Virgin Islands	VIR	2060	78.62	Americas	Americas
## 54	United States	Virgin Islands	VIR	2065	79.16	Americas	Americas
## 55	United States	Virgin Islands	VIR	2070	79.67	Americas	Americas
## 56	United States	Virgin Islands	VIR	2075	80.18	Americas	Americas
## 57	United States	Virgin Islands	VIR	2080	80.68	Americas	Americas
## 58	United States	Virgin Islands	VIR	2085	81.20	Americas	Americas
## 59	United States	Virgin Islands	VIR	2090	81.72	Americas	Americas
## 60	United States	Virgin Islands	VIR	2095	82.24	Americas	Americas

die USA existieren also doch. Übrigens: `str_detect(spalte, suchterm)` liefert für jeden Wert von `spalte`

zurück, ob sich der Suchterm (suchterm) darin befindet. Man bekommt also einen Vektor mit TRUE, FALSE, TRUE, ..." und so weiter.filter() erlaubt, so einen logischen Vektor als Grundlage des Filterns herzunehmen.

Schauen wir, wie viele Länder zu unseren Suchterm passen (also TRUE zurückliefern):

```
exp3 %>%
  filter(str_detect(Entity, "USA|United States")) %>%
  pull(Entity) %>%
  unique() # zeigt nur unterschiedliche (unique) Werte an
```

```
## [1] United States          United States Virgin Islands
## 201 Levels: Afghanistan Albania Algeria Angola ... Zimbabwe
```

Aha, "United States" und die "Virgin Islands".

Schauen wir mal nach, ob es die USA auch in der Tabelle world gibt:

```
world %>%
  select(soverecht) %>%
  filter(str_detect(soverecht, "USA|United States")) %>%
  pull(soverecht) %>%
  unique()
```

```
## [1] "United States of America"
```

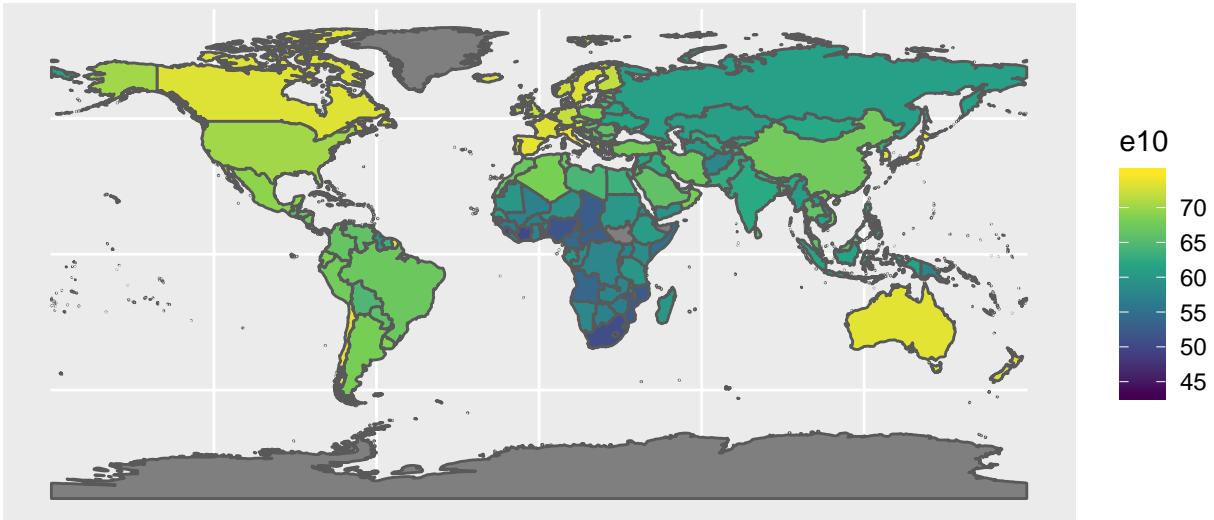
Ah - die USA heißen hier "United States of America". Das ist nicht exakt zu "United States" aus exp3. Daher hat das Verheiraten (der full_join()) oben nicht funktioniert. Besser wir nehmen den ISO-Code zum Vereinigen. Codes sind (hoffentlich) "bruchsicher", so dass das Vereinigen klappen sollte. Der relevant Code scheint in Spalte adm0_a3 zu Hause zu sein:

```
exp_2015_joined <- world %>%
  left_join(exp_2015, by = c("adm0_a3" = "Code"))
```

```
## Warning: Column `adm0_a3`/`Code` joining character vector and factor,
## coercing into character vector
```

Und Plotten:

```
gf_sf(fill = ~e10, data = exp_2015_joined) %>%
  gf_refine(scale_fill_viridis_c())
```



Hat funktioniert!