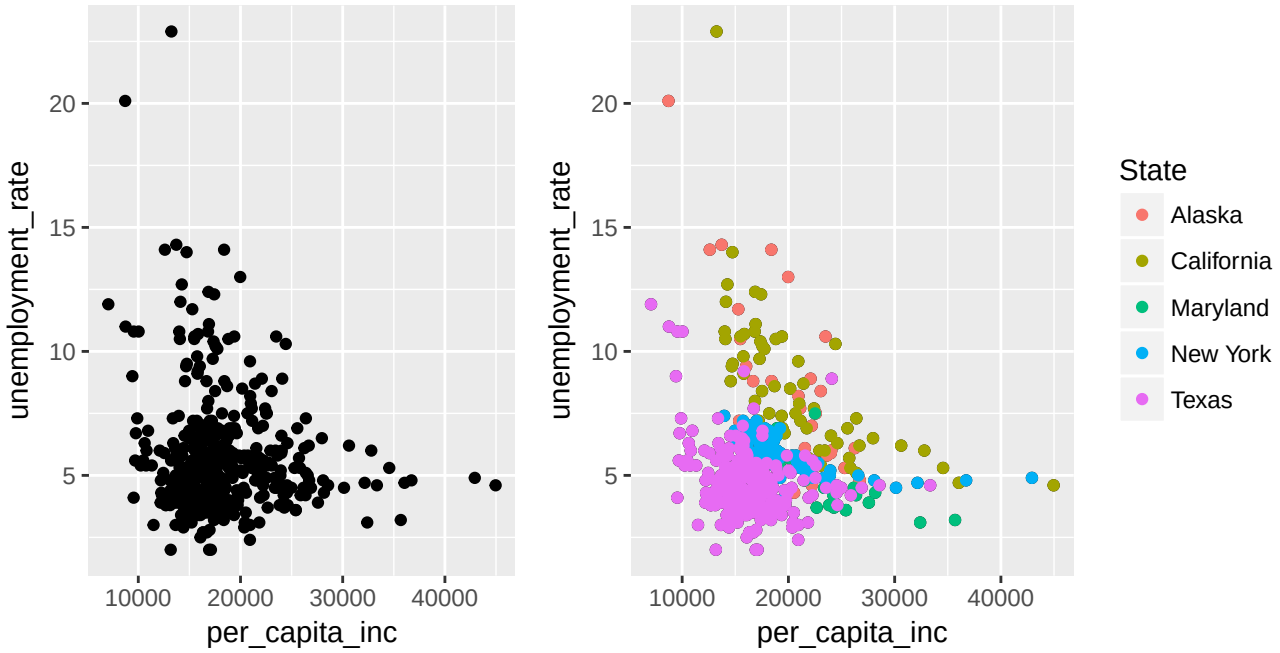
Rakendustarkvara: R. Sügis 2017, 6. praktikum*

1 Veel joonisele kihtide lisamisest

Paketiga `ggplot2` koostatud jooniseid saab salvestada objektina; valmisolevaid jooniseid saab seetõttu lihtsasti muuta ja täiendada. Eelimises praktikumis vaatasime juba hajuvusdiagrammi punktidele värvi lisamist. Teeme selle meeldetuletuseks korra veel läbi, kasutades ka võimalust joonis objektina salvestada

```
p <- ggplot(data = mk, aes(x = per_capita_inc, y = unemployment_rate)) +
            geom_point()# tekitatakse joonise objekt, ei kuvata
p # kuvatakse joonis
p + geom_point(mapping = aes(colour = State)) # lisame punktidele värvi
```



Kindlasti ei tohi unustada käsku `aes(.)` (*aesthetics*), mis aitab siduda graafilisi elemente andmestikus olevate tunnustega. Kui `aes(.)` funktsiooni ei kasutaks, otsitaks vastavate argumentide väärtuseid mitte andmestikust, vaid töökeskkonnast:

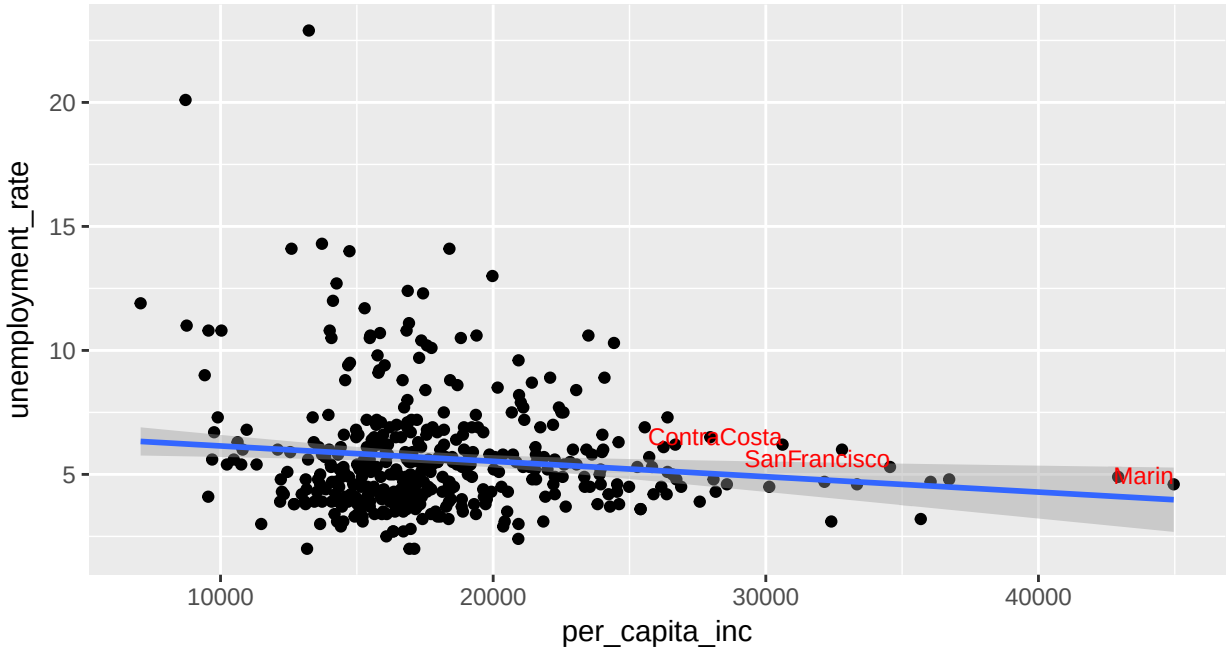
```
p + geom_point(colour = State) # objekti State otsitakse töökeskkonnast ja ei leita
```

```
## Error in layer(data = data, mapping = mapping, stat = stat, geom = GeomPoint, : object 'State' not f
```

Joonisele saab lisada ka uusi elemente, näiteks regressioonikõveraid või teksti. Uute elementide lisamisel võib ette anda uue andmestiku argumendiga `data`:

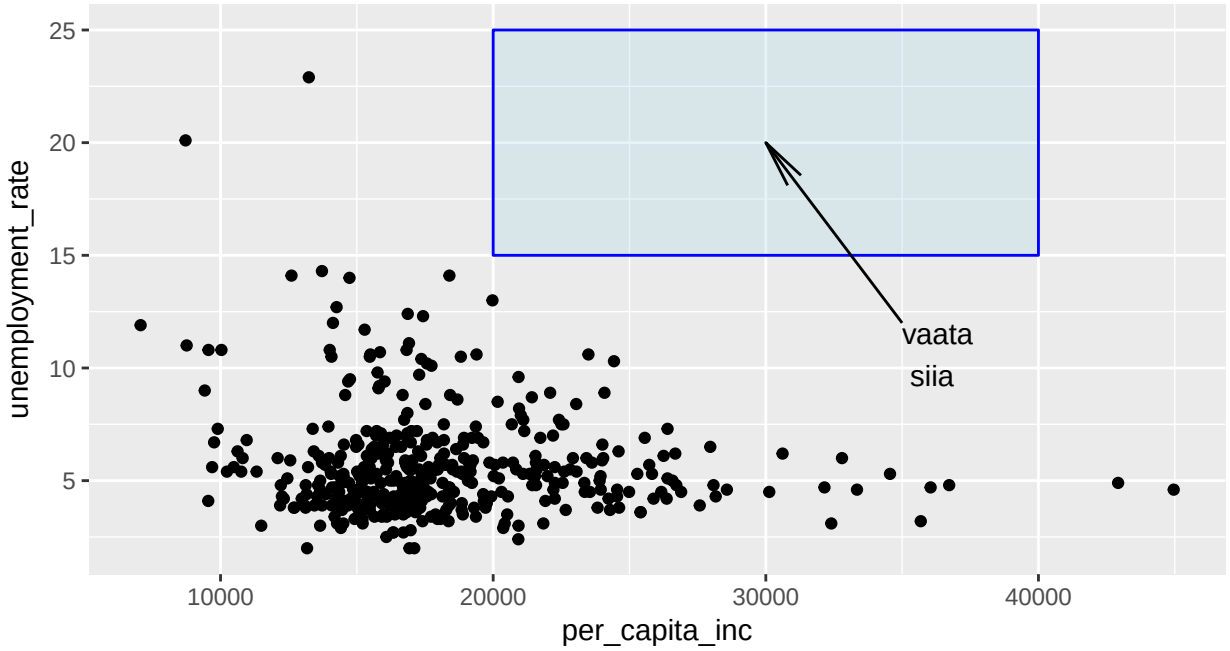
```
p + geom_smooth(method = lm) + # lisatakse siluja (praegu lineaarne regressioon)
    geom_text(data = mk[c(34, 48, 65), ], mapping = aes(label = County), size = 3,
              colour = "red", hjust = 1, vjust = 0)
```

*Praktikumijuhendid põhinevad aine MTMS.01.092 Rakendustarkvara: R (2 EAP) materjalidel, mille autorid on Mait Raag ning Raivo Kolde



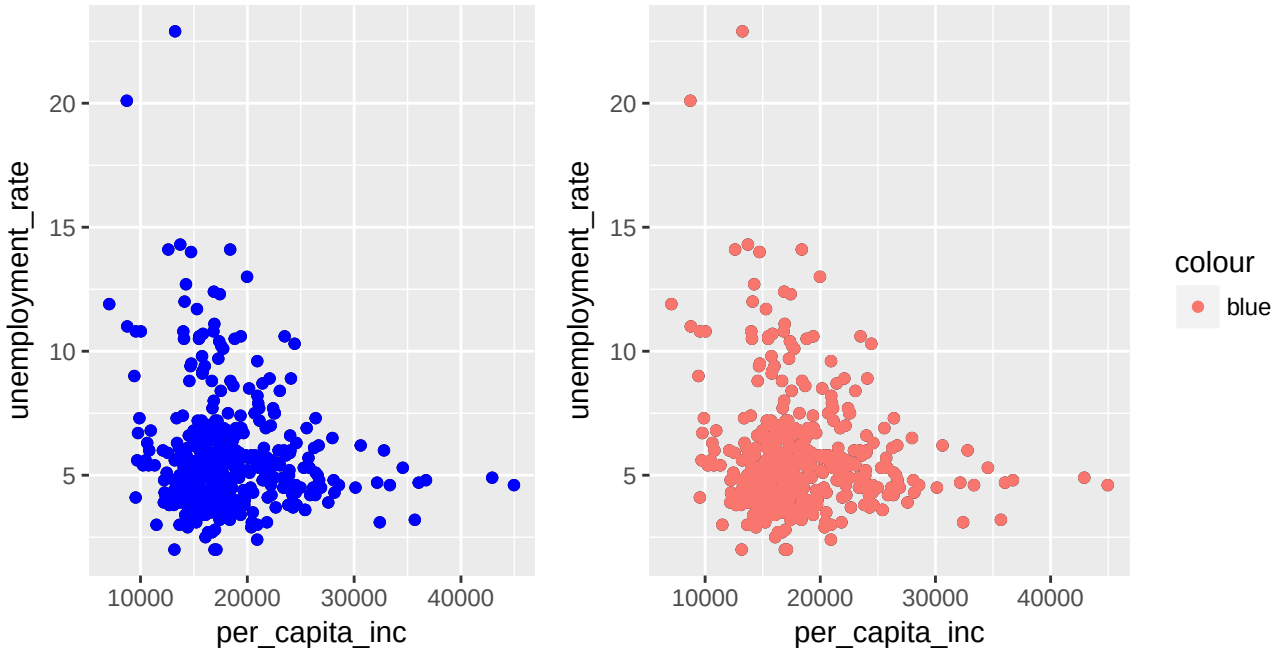
Kui joonisele on vaja lisada mingeid üksikuid detaile või märkusi, mis ei tulene enam otseselt kasutatud andmestikust, siis saab seda teha käsu `annotate(.)` abil:

```
p +
  annotate("rect", xmin = 2*10^4, xmax = 4*10^4, ymin = 15, ymax = 25,
           fill = "lightblue", alpha = 0.3, colour = "blue") +
  annotate("segment", x = 3.5*10^4, xend = 3*10^4, y = 12, yend = 20,
           arrow = arrow(ends = "last", angle = 10)) +
  annotate("text", x = 3.5*10^4, y = 12, label = "vaata\n siia", hjust = 0, vjust = 1)
```



Eelnevalt oli mainitud, et tunnused, mida tahame joonisel graafiliste elementidega siduda tuleks esitada läbi `aes()` käsu. Juhul, kui tahame näiteks hajuvusdiagrammi punktide värvi määrata mitte tunnuse põhjal muutuvana vaid hoopis fikseerida, siis tuleks värvi argument just `aes(.)` käsust välja jätta, sest vastasel juhul võib tulemus vahel ootamatu olla. Lisaks lisatakse joonisele legend, mis ühe värvi fikseerimise korral mõtet ei oma. Võrdle järgmist koodi ja tulemusi:

```
p + geom_point(color = "blue")
p + geom_point(aes(colour = "blue"))
```



1.0.1 Ülesanded

1. Loe sisse andmestik: `read.table("http://kodu.ut.ee/~annes/Rkursus/maakonnad.txt", sep = " ", header=T)`
2. Joonista hajuvusdiagramm `high_scl` ja `bachelor` vahel.
3. Lisa neile maakondadele nimed, kus keskkooliharidusega inimeste osakaal on $< 50\%$.
4. Lisa graafikule vertikaaljoon kohale, kus keskkooliharidusega inimeste osakaal oleks täpselt 50% (vihje: kasuta käsku `geom_vline(.)`)
5. Lisa viimasele graafikule veel oma valitud kohta mingi tekstikommentaar.

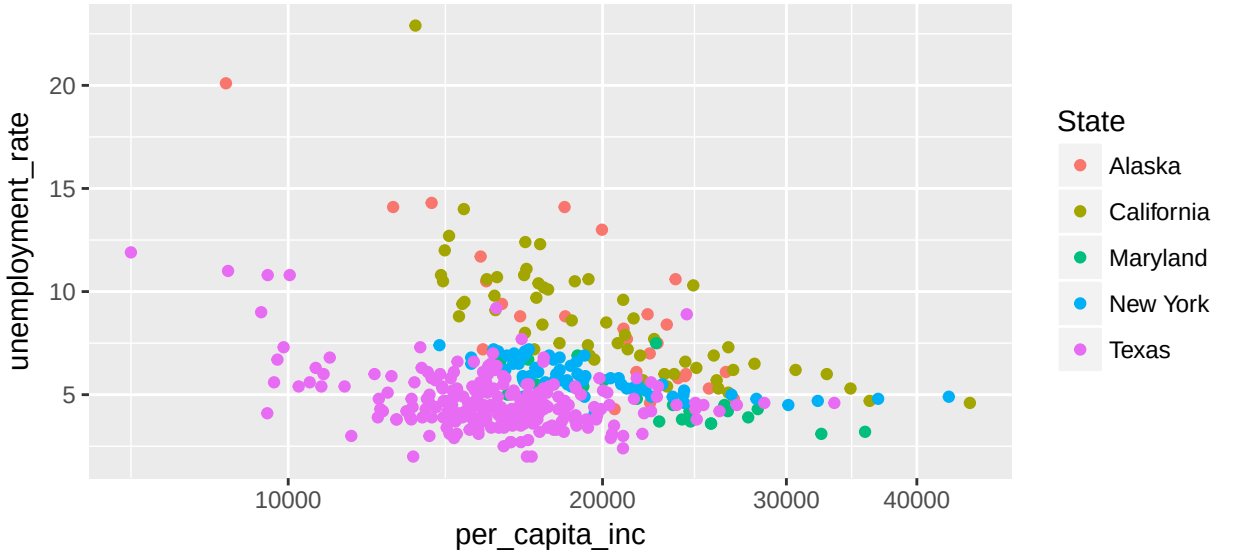
2 Skaalade muutmine

Kui `ggplot2`-ga koostada joonis, siis muutujad seostatakse graafiliste elementide omadustega (asukoht x-teljel, asukoht y-teljel, värvus, suurus jne). Vaikimisi kasutatakse küllalt mõistlikke skaalasid, nt värviskaalad on valitud selliselt, et kõik värvid oleks võrdse intensiivsusega. Mõnikord on aga soov skaalasid muuta. Selleks saab kasutada käske `scale_<graafilise elemendi omadus>_<skaala nimi>`. Näiteks `scale_x_continuous(.)` käsuga saab muuta x-telje vahemikku, `scale_colour_grey(.)` muudab värviskaala mustvalgeks.

Kõiki skaalade muutmise funktsioone on võimalik näha ggplot2 kodulehel http://ggplot2.tidyverse.org/reference/ jaotuse "Scales" all.

Skaleerimisfunktsiooni rakendamiseks tuleb see lisada joonisele:

```
p2 <- ggplot(data = mk, aes(per_capita_inc, unemployment_rate, colour = State)) + geom_point()
p2 + scale_x_continuous(trans = "log10", breaks = c(1/2, 1:4) * 10^4)
```

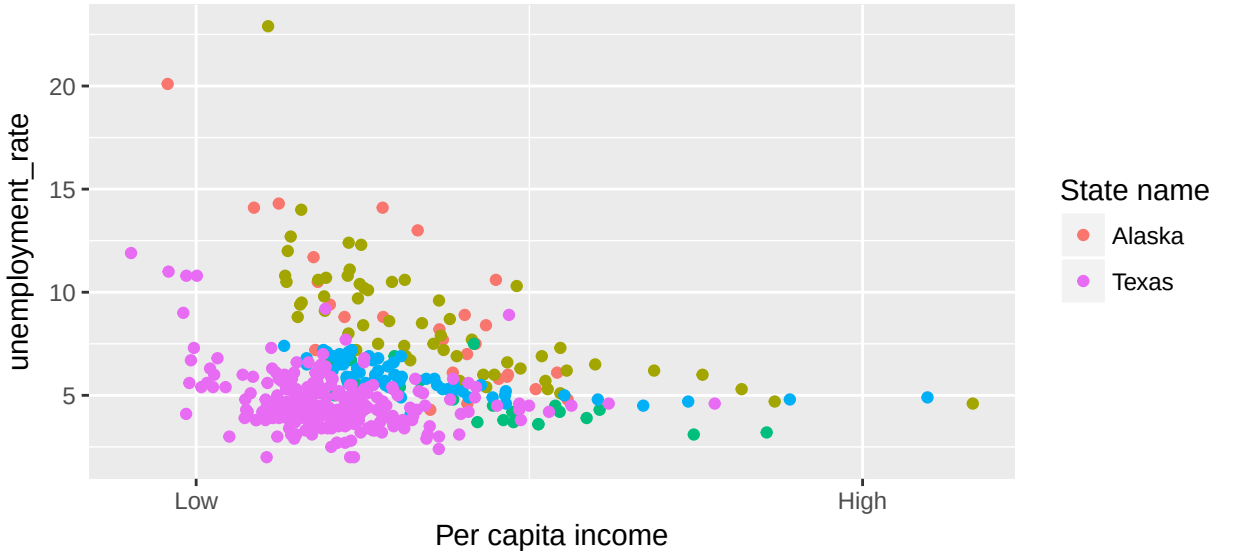


Eelneva käsuga lisati x-telje skaala muudatus: telg kuvatakse logaritmilisel skaalal, `breaks` määrab teljel esitatavad arvväärtused.

Iga erineva omaduse skaala muutmiseks saab joonisele "juurde liita" uue funktsiooni. Konkreetse funktsiooni parameetrid sõltuvad omaduse tüübist (nt punkti kuju ei saa logaritmida), aga kõigil skaleerimisfunktsioonidel on kindlasti kolm argumenti:

- `name` – telgede puhul telje nimi; värvide, kuju jm legendis antava info puhul vastava legendi pealkiri
- `breaks` – vektor punktidega, mis määravad, millised väärtused joonisel ära markeeritakse (nt x-telje jaotis)
- `labels` – parameetriga `breaks` määratud punktidele vastavad sildid

```
p2  + scale_x_continuous(name = "Per capita income", breaks = c(10000, 40000),
                         labels = c("Low", "High")) +
      scale_colour_hue(name = "State name", breaks = c("Alaska", "Texas"))
```

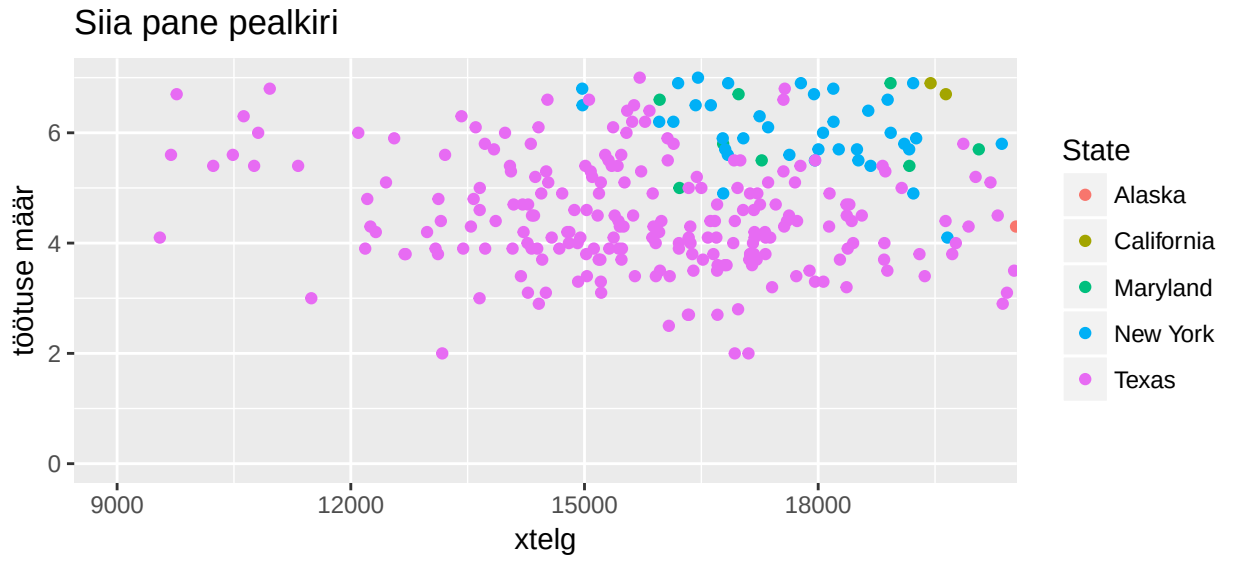


x-ja y-telgede skaalasid (mh nende nimesid) saab muuta vastava `scale`-käsuga, ent on veel võimalusi lisada telgedele nimed ja piiritleda väärtusvahemikku:

- `xlab(.)`, `ylab(.)` – vastavate telgede pealkirjad
- `labs(.)` – saab määrata nii telgede kui graafiku pealkirja (`x = "..."`, `y = "..."`, `title = "..."`)
- `xlim(.)`, `ylim(.)` – argumendiks kaks väärtust, mis määravad telgede väärtusvahemiku. NB! objektid, mille väärtused on väljaspool määratud vahemikku asendatakse `NA` väärtusega ja jäetakse välja kõigilt kihtidelt.
- `coord_cartesian(.)` – argumentidega `xlim`, `ylim` saab määrata xy-teljestikus väärtusvahemikud, mis nähtavale jäävad. See on võimalus joonise mingile piirkonnale 'suumida'.
- `ggtitle(.)` – joonise pealkiri

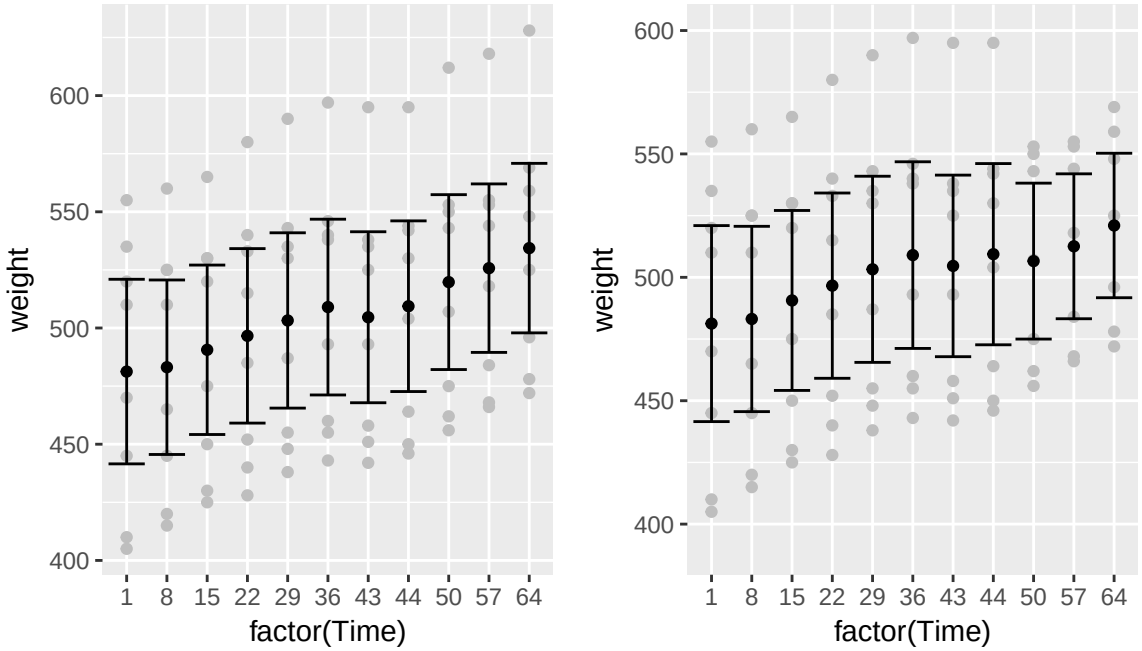
Näiteks joonise pealkirja ja teljetiitlite muutmine ning telgede väärtusvahemiku määramine y-teljel andmetest väärtusvahemiku valimise teel, x-teljel joonisest lõike tegemise teel:

```
p2 + ylab("töötuse määr") + labs(title = "Siia pane pealkiri", x = "xtelg" ) +
        ylim(0, 7) + coord_cartesian(xlim = c(9000, 20000))
```



Teljepiiride määramine `ylim(.)` või `xlim(.)` käsuga võib mõjutada joonisel esitatavaid arvutuslikke tulemusi. Näiteks kui joonisel esitada keskväärtused koos usalduspiiridega ning määrata teljepiirid nii, et mõni arvutuse aluseks olev vaatlus jääb piiridest välja, siis muutuvad ka usalduspiirid. Näide sellisest olukorrast on järgmisel joonisel (siin on kasutuses on teine näiteandmestik: rottide kehakaalu ja dieetide andmestik)

```
rotid <- nlme::BodyWeight
rotid <- rotid[rotid$Diet != 1, ]
p <- ggplot(rotid, aes(factor(Time), weight)) + geom_point(color = "gray") +
  stat_summary(geom = "errorbar", fun.data = mean_se, fun.args = list(mult = 1.96)) +
  stat_summary(geom = "point", fun.y = mean)
p
p + ylim(390, 600)
```




```
## Warning: Removed 3 rows containing non-finite values (stat_summary).
```

```
## Warning: Removed 3 rows containing non-finite values (stat_summary).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

Kuna kolm vaatlust andmetes on üle 600, siis määratud piir `ylim(390, 600)` jätab need välja kõigilt kihtidelt. Väljundisse tuleb kolm hoiatust, üks hoiatus iga lisatava kihi kohta (hajuvusdiagramm, usalduspiirid ja keskväärtus punktina kiht).

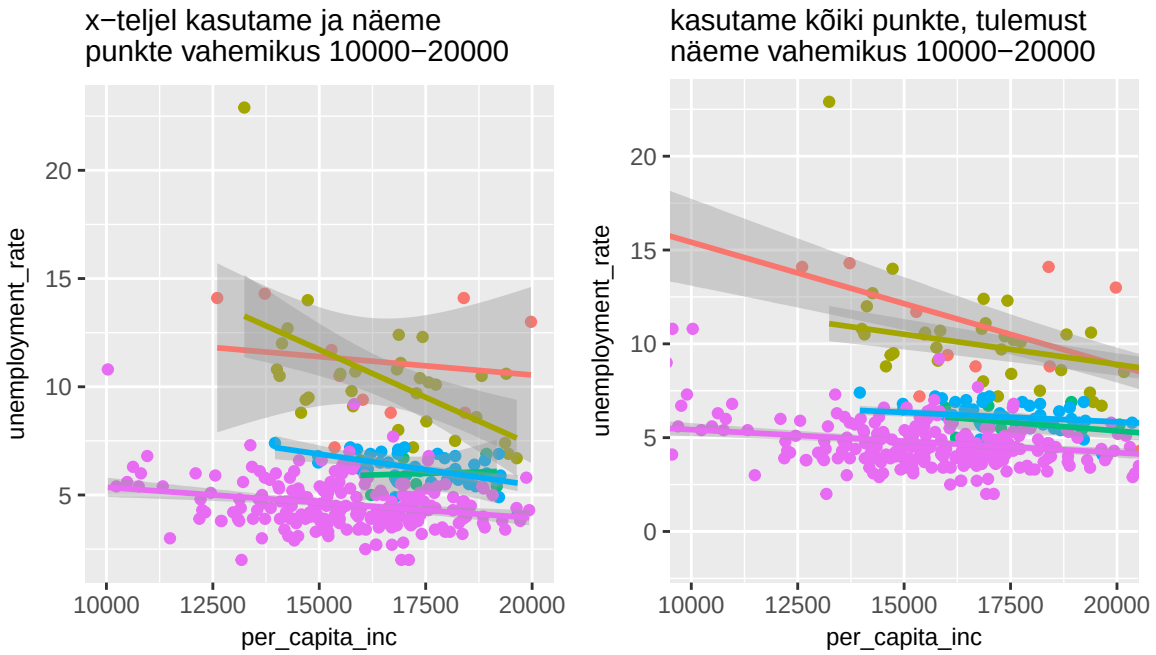
2.1 Pidevate skaalade muutmine

Pidevate skaalade muutmise käskudel (nt `scale_<?>_continuous`, `scale_<?>_gradient`) on mõned spetsifilised argumendid:

- `trans` – skaala transformeerimise funktsiooni nimi, nt "exp", "log", "log10", "sqrt".
- `limits` – kahe-elemendiline vektor, mis annab skaala algus- ja lõpp-punkti. Sarnalset `xlim(.)`, `ylim(.)` käskudega tuleb selle argumendi kasutamisel olla tähelepanelik, sest andmed, mis vastavast vahemikust välja jäävad asendatakse `NA` väärtustega. Selle toimingu mõju sõltub muudetavast skaalast: x-ja y-telgede piiride määramisel tähendab see, et neid andmeridu ei kasutata joonise tegemisel (nt ka regressioonisirge arvutamisel). Värviskaala korral määrab see objektid, millele värvi ei määrata(vt alapunkt "Värviskaala muutmine").

Argumendi `limits` mõju vaata järgmistel joonistel regressioonsirgeid võrreldes.

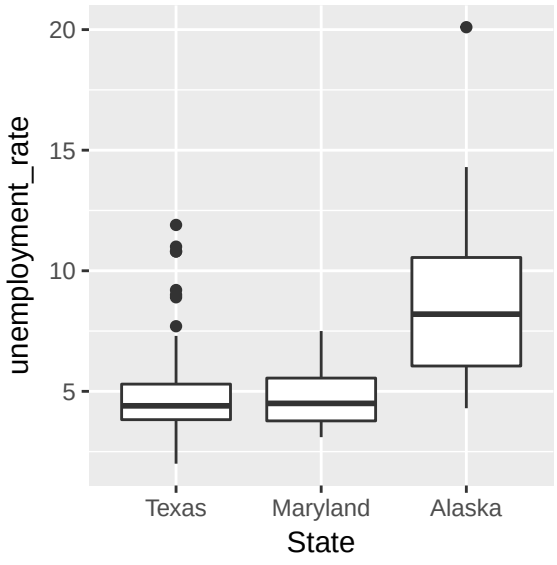
```
p2 + scale_x_continuous(limits = 1:2*10^4) + geom_smooth(method = lm) +
  labs(title = "x-teljel kasutame ja näeme\npunkte vahemikus 10000-20000")
p2 + coord_cartesian(xlim = 1:2*10^4) + geom_smooth(method = lm) +
  labs(title = "kasutame kõiki punkte, tulemust\nnäeme vahemikus 10000-20000")
```



2.2 Diskreetsete skaalade muutmine

Diskreetsetel skaaladel töötab argument `limits` teistmoodi: nimelt saab sellega ette anda konkreetsed väärtused, mida joonisel kujutatakse, ülejäänud väärtusi siis joonisel ei kujutata. Oluline on ka väärtuste etteandmise järjekord:

```
ggplot(data = mk, aes(State, unemployment_rate)) + geom_boxplot() +
  scale_x_discrete(limits = c("Texas", "Maryland", "Alaska"))
```



2.2.1 Ülesanded

1. Pane eelmises ülesandes tehtud joonisel y-telje nimeks "Higher education percentage"; muuda telje vahemikku (0, 100); muuda teljel olevaid silte ja nende paigutust nii, et need oleksid kujul 0%, 25%, 50%, 75%, 100%.
2. Jaga tunnuse `per_capita_inc` väärtused viide vahemikku nii, et igas vahemikus oleks üks viiendik vaatlustest. Pane uuele tunnusele nimi `income_class` ja sildista tekkinud väärtusklassid *"Very low", "Low", "Medium", "High", "Very high"*.
3. Lisa pildile värviga tunnus `income_class` nii, et värvitaks ainult punkte, kus sissetulek on kas väga kõrge või väga madal.

2.3 Värviskaala muutmine

Värvide valik joonisel on väga oluline. Õigesti valitud värvidega on võimalik tuua selgemini välja oma sõnumit, muuta joonist loetavamaks ja meeldivamaks. Asjakohane värvus sõltub tunnusest, mida soovitakse kujutada. Üldiselt võib värviskaalad jaotada kolmeks:

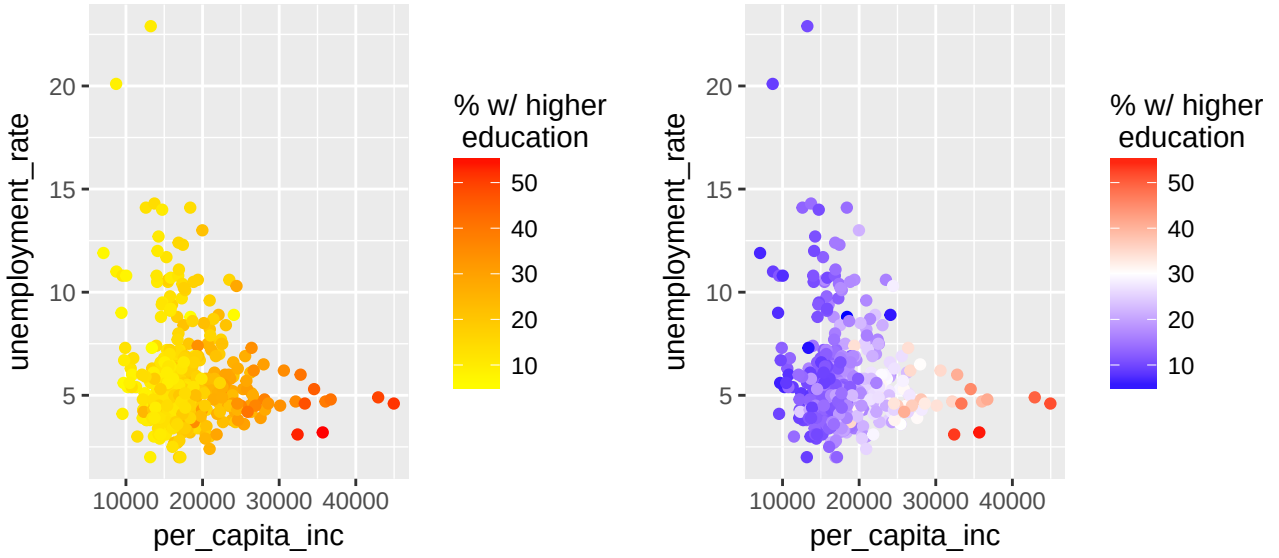
- gradient – pidevate tunnuste jaoks; kõige väiksem ja kõige suurem väärtus vastavad mingitele värvidele ning vahepealsed väärtused nende kahe värvi segule
- lahknev gradient – pidevate tunnuste jaoks, kui pideval tunnusel on mingi selge nullpunkt (nt õhutemperatuur, tsentreeritud skoor vms); kaks ekstreemset väärtust ja nullpunkt vastavad mingile puhtale värvile, vahepealsed väärtused kahe värvi segule.
- kvalitatiivne – diskreetsete tunnuste jaoks; iga väärtuse jaoks kasutatakse võimalikult erinevat värvitooni. Samas on oluline silmas pidada, et heleduselt ja intensiivsuselt oleks kõik värvid võrdsed.

Gradientskaalat saab kontrollida käsuga `scale_<?>_gradient(.)` (küsimärgi asemel on tavaliselt `fill` või `colour`). Saab kasutada kõiki pideva tunnuse skaala muutmise argumente, ning lisaks on kaks argumenti: `low` ja `high` ekstreemsete väärtuste värvi määramiseks gradiendil. Lahknevat gradienti saab kontrollida funktsiooniga `scale_<?>_gradient2(.)`, millel on lisaks `low` ja `high` väärtustele argument `mid` millega saab ette värvi nime, mis vastab nullpunktile, vaikimisi on see värv valge. Skaala nullpunkti väärtuse saab ka ise ette anda. Lahknevat ja veel keerulisemaid gradiente saab määrata funktsiooniga `scale_<?>_gradientn`, mille argumendile `colours` saab ette anda vektori värvidega, mille vahele siis uued värvid sujuva üleminekuga valitakse.

Näited kahe gradientskaala käsu kasutamisest:

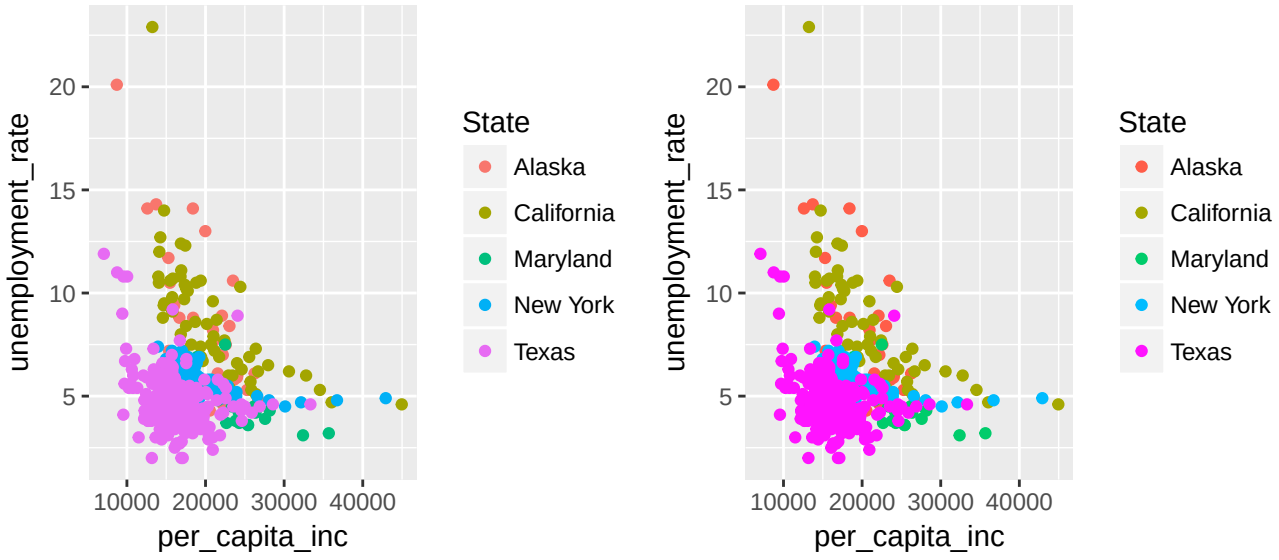
```
p3 <- ggplot(data = mk, aes(per_capita_inc, unemployment_rate)) +
  geom_point(aes(colour = bachelor ))

nimi = "% w/ higher \n education"
p3 + scale_colour_gradient(name = nimi, low = "yellow", high = "red")
p3 + scale_colour_gradient2(name = nimi, low = "blue", high = "red", midpoint = 30)
```



Diskreetsete/kvalitatiivsete värviskaalade kontrollimiseks kasutatakse vaikimisi funktisooni `scale_<?>_hue(.)`, mis valib HCL värviskaalal[1] parameetri `h` (*hue* värvitoon) väärtused võimalikult erinevad, jättes värvi tugevuse `c` (*chroma*) ja heleduse `l`(*luminance, lightness*) konstantseks. Nii saadakse võimalikult erinevad värvid, mis samal ajal on ühesuguse intensiivsusega.

```
p4 <- ggplot(data = mk, aes(per_capita_inc, unemployment_rate)) + geom_point(aes(colour = State ))
p4
p4 + scale_colour_hue(c = 150) # tugevus suuremaks, vaikimisi 100
#p4 + scale_colour_hue(l = 20) # heledus madalamaks, vaikimisi 65
#p4 + scale_colour_hue(h = c(10, 190)) # kaks värvitooni, skaala algus- ja lõpptoon
```

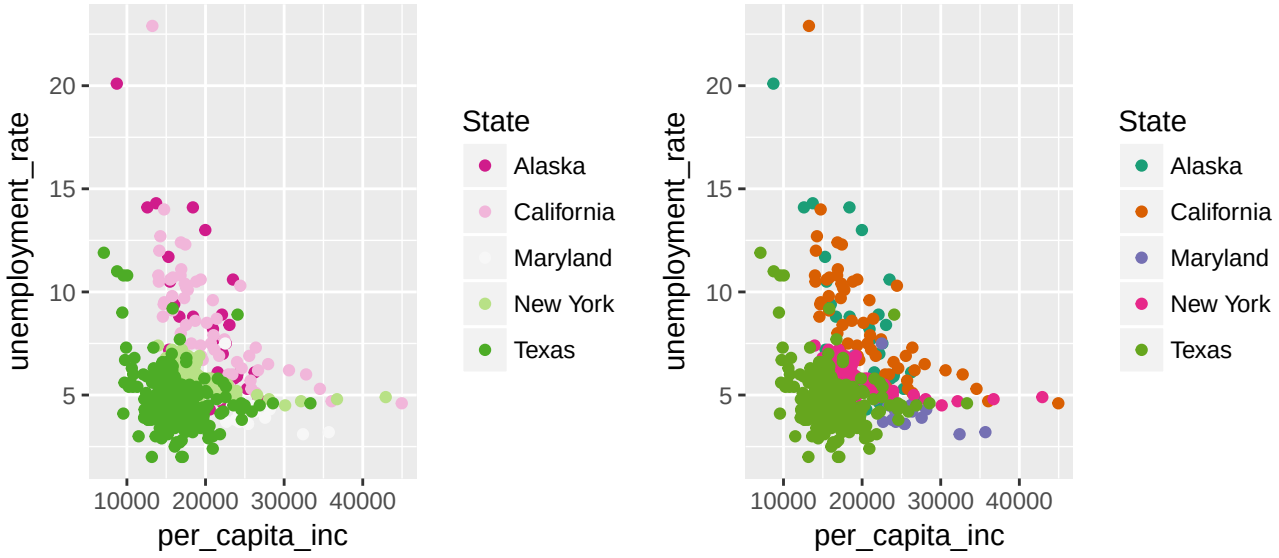


[1] http://en.wikipedia.org/wiki/Munsell_color_system

Diskreetsete värviskaalade jaoks on arendatud värvipalett "Colorbrewer"[2] (algselt arendatud maakaartide värvimiseks). `ggplot2`-s pääseb sellele paletile ligi funktsiooniga `scale_<?>_brewer(.)`, millel on kaks argumenti:

- `type` – võimalikud väärtused on `"seq"`, `"div"` ja `"qual"`
- `palette` – paleti number (vt http://www.colorbrewer2.org)

```
p4 + scale_colour_brewer(type = "div", palette = 2)
p4 + scale_colour_brewer(type = "qual", palette = 2)
```



2.3.1 Ülesanded

1. Proovi eelmises ülesandes tehtud joonisel (`high_scl` ja `bachelor`) valida `income_class` jaoks erinevaid värviskaalasid.

[2] http://www.colorbrewer2.org

3 Joonise viimistlemine

Vaikimisi joonistab `ggplot2` halli taustaga jooniseid – et heledamad värvid oleksid sama silmatorkavad kui tumedad. Kui on soov valge tausta järele, siis seda saab tellida, lisades joonisele käsu `+ theme_bw()`. Veelgi detailsemalt saab joonise kujundust käsuga `+ theme(.)`, mille argumentidest mõned olulisemad on:

Argument	Elemendi tüüp	Selgitus
line	line	kõik jooned
rect	rect	kõik ristkülikulised elemendid (taustad, raamid)
text	text	kõik tekstid
axis.line	line	joonise teljed
axis.text.x	text	x-telje väärtused
axis.text.y	text	y-telje väärtused
axis.ticks	line	joonise jaotised
axis.title.x	text	x-telje pealkiri
axis.title.y	text	y-telje pealkiri
legend.background	rect	legendi taust
legend.key	rect	legendi võtme taust
legend.text	text	legendi tekst
legend.title	text	legendi pealkiri
panel.background	rect	graafiku taust
panel.border	rect	graafikut ümbritsev raam
panel.grid.major	line	jaotise jooned
panel.grid.minor	line	jaotise jooned
plot.background	rect	kogu joonise taust
plot.title	text	joonise pealkiri
strip.background	rect	tahkude pealkirjade taust
strip.text.x	text	horisontaalsete tahkude pealkiri
strip.text.y	text	vertikaalsete tahkude pealkiri

Täielikum loetelu `theme(.)` võimalikest argumentidest on aadressil http://ggplot2.tidyverse.org/reference/#section-themes .

Nagu näha, on kolm peamist elemenditüüpi, mida saab muuta käskudega `element_text(.)`, `element_line(.)` ja `element_rect(.)`. Nendel käskudel on omakorda argumendid:

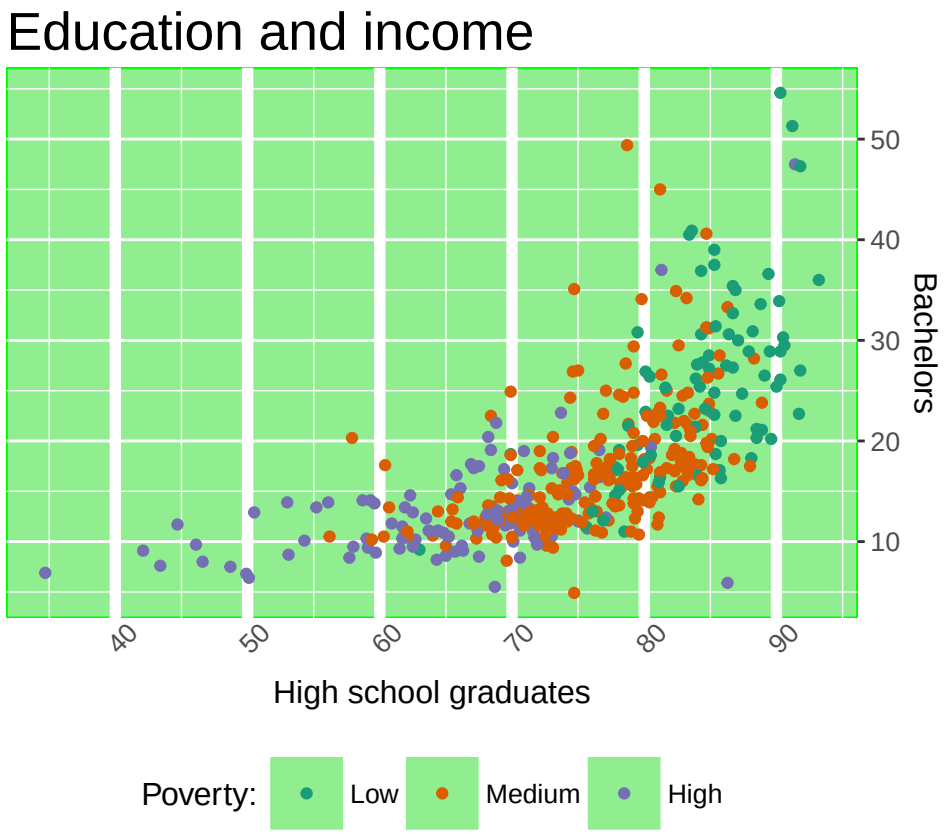
- `element_text(.)` – `family`, `size`, `colour`, `angle`, `hjust`, `vjust` kontrollivad teksti šrifti, suurust, värvi, kaldenurka ja positsiooni vaikimisi määratud koha suhtes. Näiteks `size=14` valib teksti punktisuuruseks 14pt.

- `element_line(.)` – `colour`, `size`, `linetype`;
- `element_rect(.)` – `colour`, `size`, `linetype` ääre muutmiseks, `fill` sisemuse värvi muutmiseks

Kui mõnda elementi üldse joonisel ei soovi näha, siis tuleb vastavale argumendile anda väärtus `element_blank()`.

3.0.1 Ülesanded

1. Tee allolevaga võimalikult sarnane joonis.



4 Joonise salvestamine

ggplot2-ga tehtud jooniseid saab salvestada käsuga `ggsave(.)`. Kui anda käsule vaid faili nimi (koos laiendiga), siis salvestatakse viimati valmistatud joonis. Kui argumente `widht` ja `height` ei kasuta, siis joonise mõõtmed võetakse joonise akna järgi.