

Cancer_reg_EDA

PCA group

2022-07-14

```
# load packages
library(dplyr)

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

library(ggplot2)
library(readr)
library(haven)
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v tibble  3.1.7      v stringr 1.4.0
## v tidyr   1.2.0      v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(sandwich)
library(lmtest)

## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric

library(stargazer)

##
## Please cite as:
## Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
## R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```

library(Hmisc)

## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
## The following objects are masked from 'package:base':
##
##     format.pval, units
library(funModeling)

## funModeling v.1.9.4 :)
## Examples and tutorials at livebook.datascienceheroes.com
## / Now in Spanish: librovivodecienciadedatos.ai
library(olsrr)

##
## Attaching package: 'olsrr'
## The following object is masked from 'package:datasets':
##
##     rivers
cancer <- read_csv("cancer_reg.csv")

## Rows: 3047 Columns: 33
## -- Column specification -----
## Delimiter: ","
## chr (2): binnedinc, geography
## dbl (31): avganncount, avgdeathsperyear, target_deathrate, incidencerate, me...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

```

Correlation table:

```

correlation_table(data = cancer, target = "target_deathrate")

##           Variable target_deathrate
## 1      target_deathrate           1.00
## 2  pctpubliccoveragealone           0.39
## 3      incidencerate              0.38
## 4      pcths25_over              0.38
## 5      povertypercent              0.37
## 6      pctpubliccoverage           0.35
## 7  pctunemployed16_over           0.32
## 8      pctblack                   0.26

```

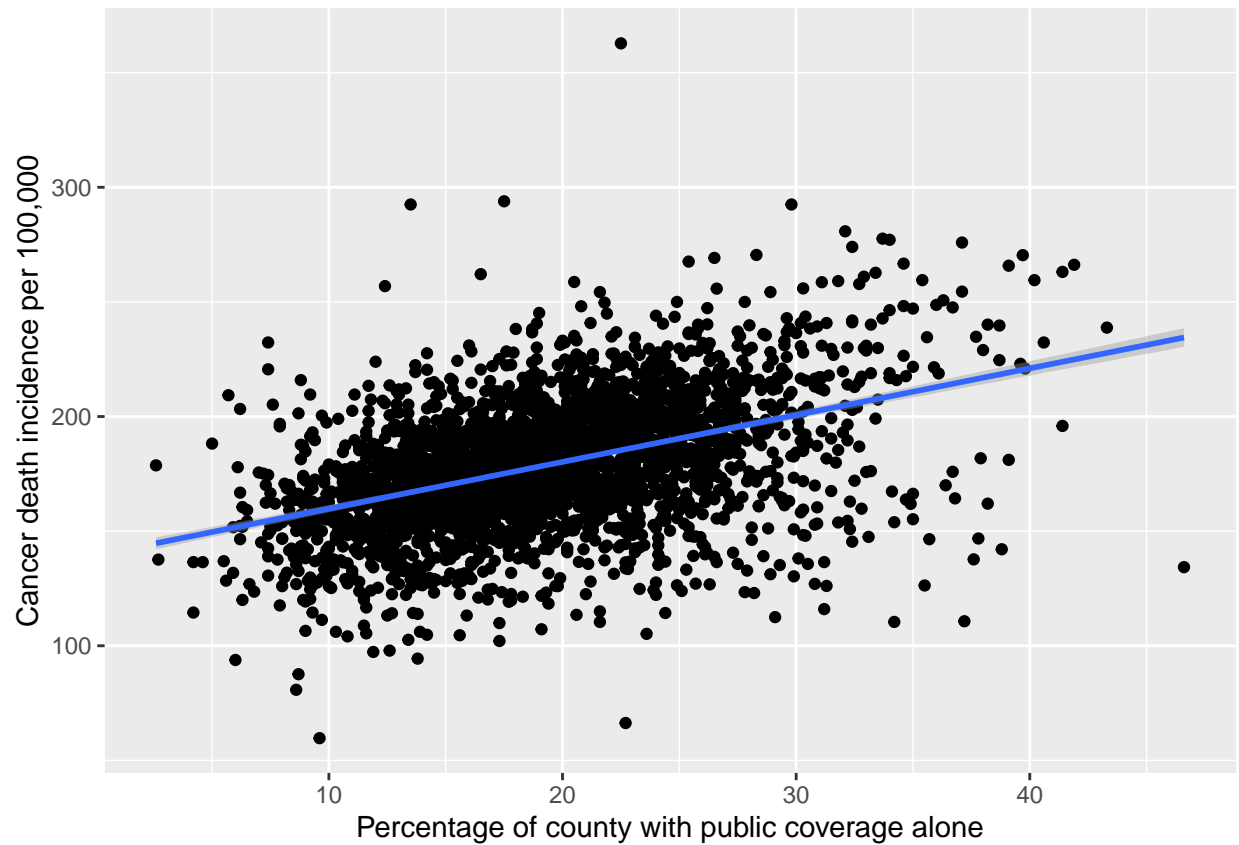
## 9	pcths18_24	0.25
## 10	pctnohs18_24	0.08
## 11	medianagefemale	0.02
## 12	medianage	-0.01
## 13	studypercap	-0.03
## 14	medianagemale	-0.03
## 15	birthrate	-0.05
## 16	avgdeathsperyear	-0.10
## 17	popest2015	-0.11
## 18	avganncount	-0.13
## 19	pctsomecol18_24	-0.16
## 20	pctwhite	-0.16
## 21	pctasian	-0.20
## 22	pctotherrace	-0.21
## 23	percentmarried	-0.23
## 24	pctempprivcoverage	-0.23
## 25	pctmarriedhouseholds	-0.29
## 26	pctbachdeg18_24	-0.30
## 27	pctprivatecoveragealone	-0.32
## 28	pctprivatecoverage	-0.34
## 29	pctemployed16_over	-0.37
## 30	medincome	-0.38
## 31	pctbachdeg25_over	-0.44

Plot positive correlations:

Positive correlations: pctpubliccoveragealone, incidencerate, pcths25_over, povertypercent, pctpubliccoverage, pctunemployed16_over

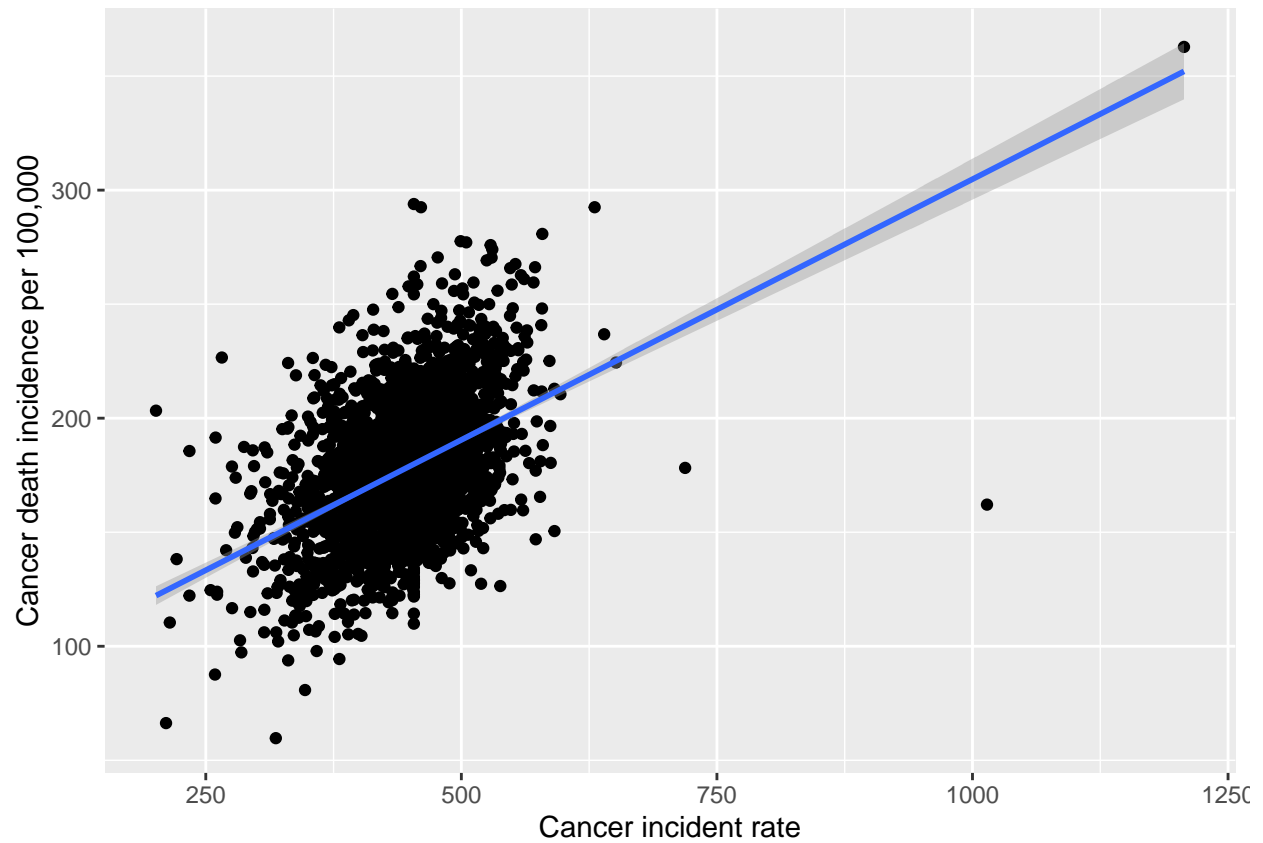
```
ggplot(data = cancer, aes(x = pctpubliccoveragealone, y = target_deathrate))+
  geom_point()+
  geom_smooth(method = lm)+labs(x = "Percentage of county with public coverage alone", y = "Cancer death rate")

## `geom_smooth()` using formula 'y ~ x'
```



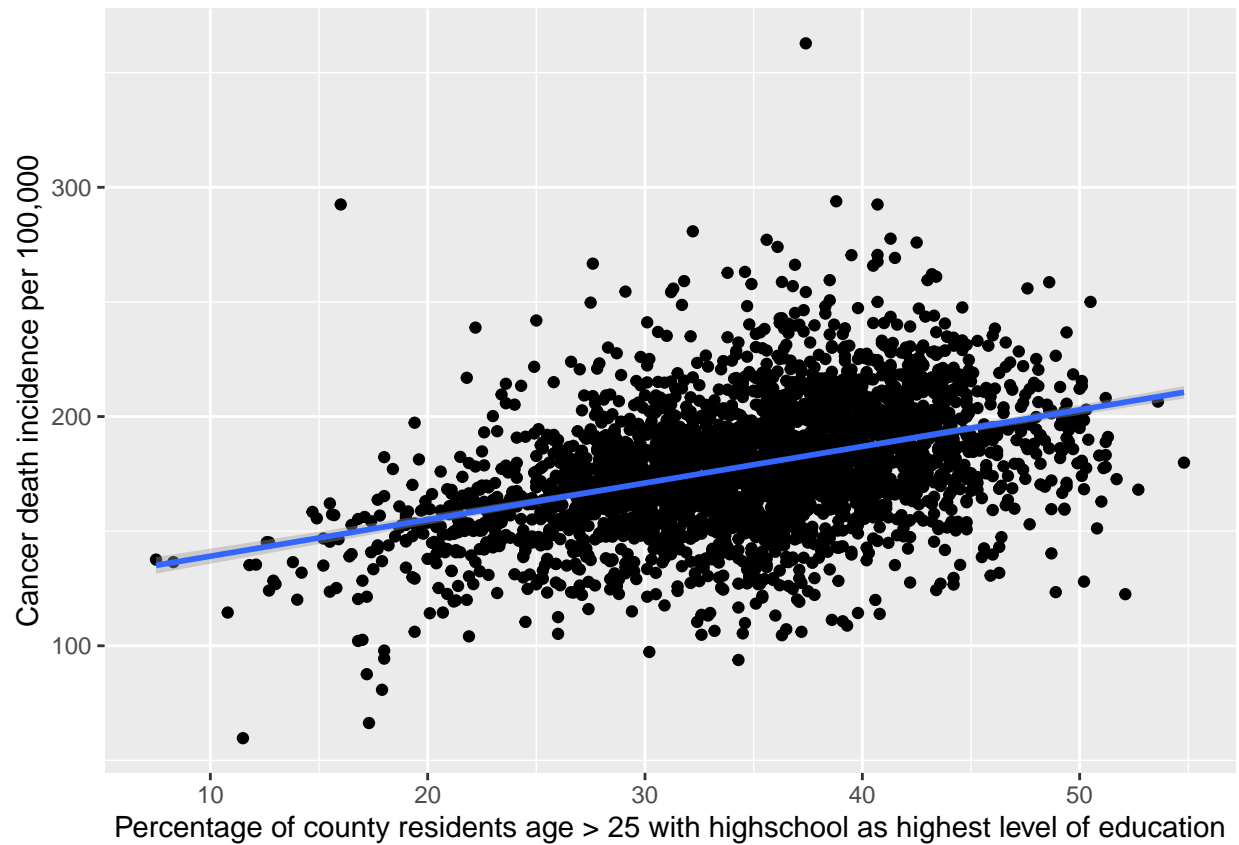
```
ggplot(data = cancer, aes(x = incidencerate, y = target_deathrate))+
  geom_point()+
  geom_smooth(method = lm)+labs(x = "Cancer incident rate", y = "Cancer death incidence per 100,000")

## `geom_smooth()` using formula 'y ~ x'
```



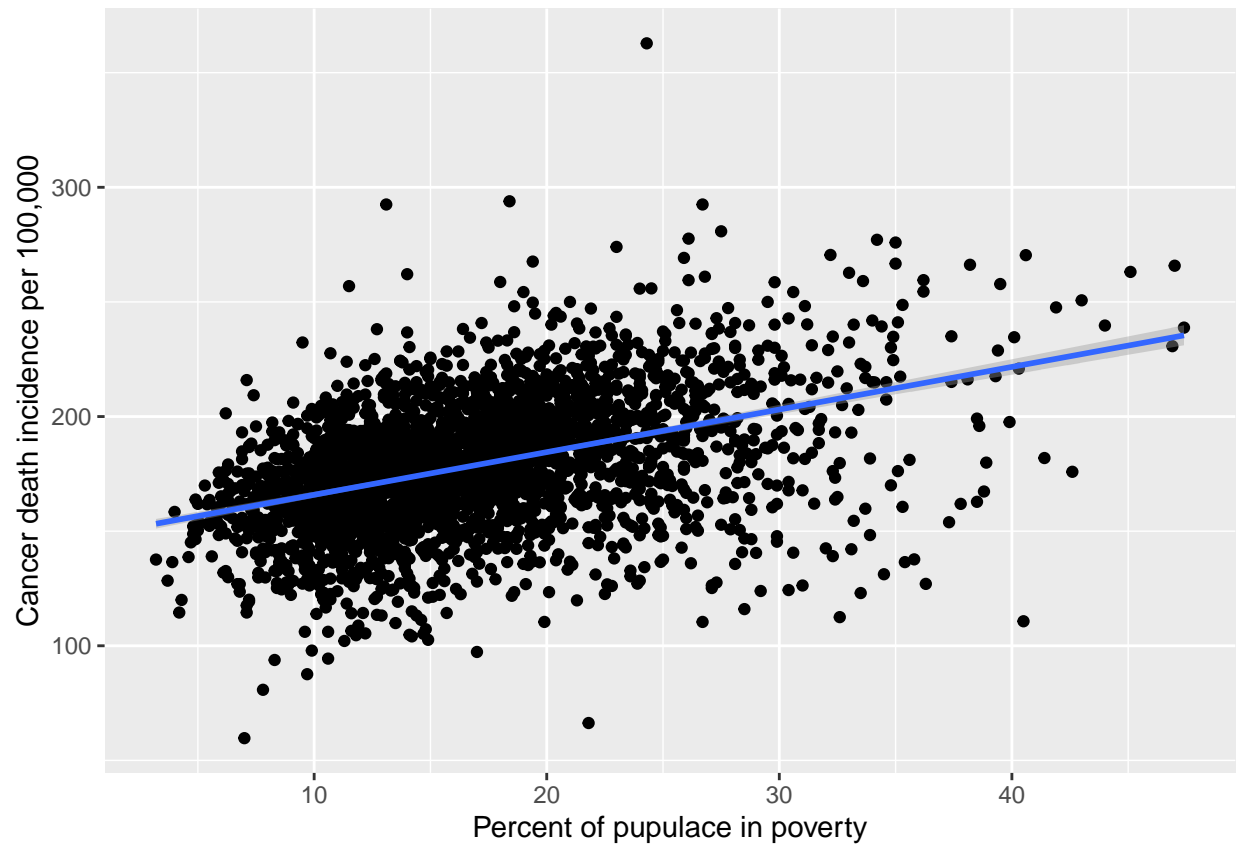
```
ggplot(data = cancer, aes(x = pcths25_over, y = target_deathrate))+
  geom_point()+
  geom_smooth(method = lm)+labs(x="Percentage of county residents age > 25 with highschool as highest l

## `geom_smooth()` using formula 'y ~ x'
```



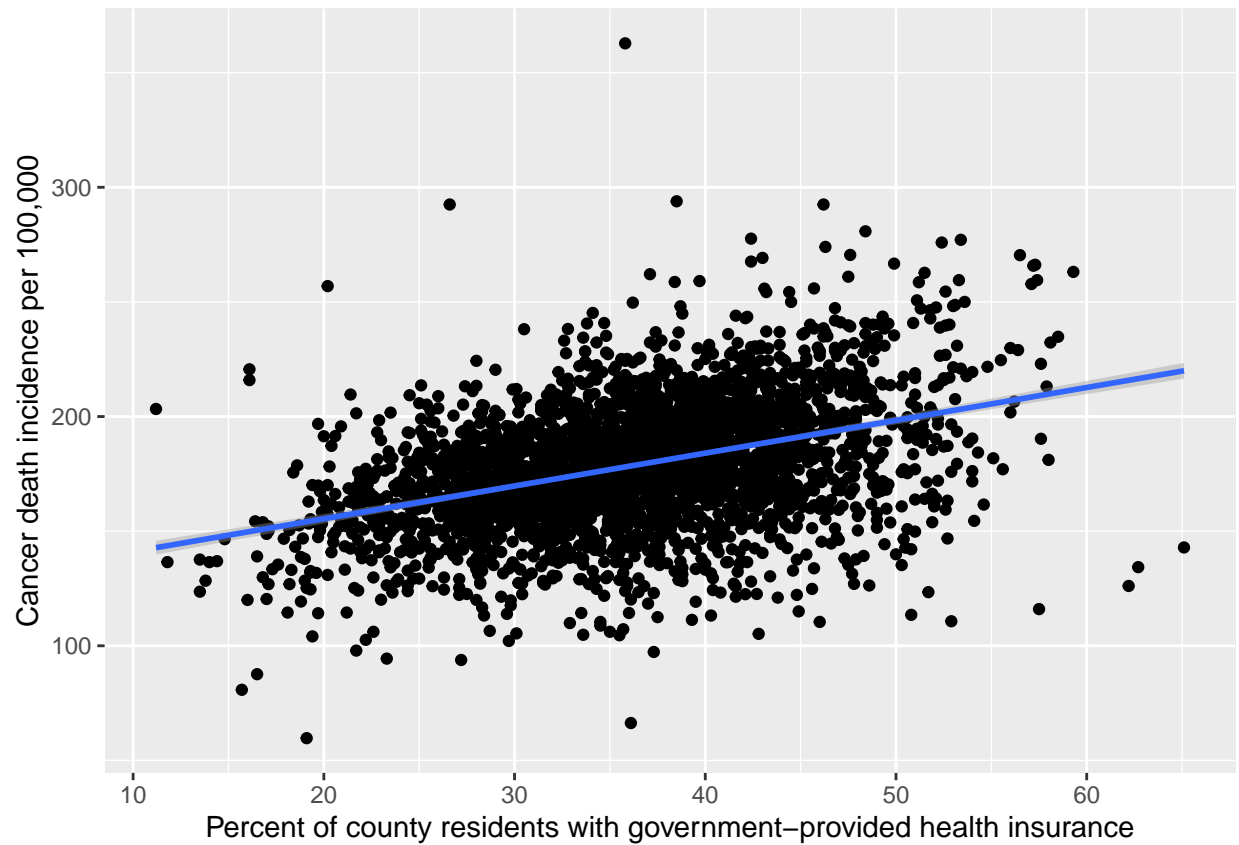
```
ggplot(data = cancer, aes(x = povertypercent, y = target_deathrate))+
  geom_point()+
  geom_smooth(method = lm)+
  labs(x="Percent of pupulace in poverty", y= "Cancer death incidence per 100,000")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



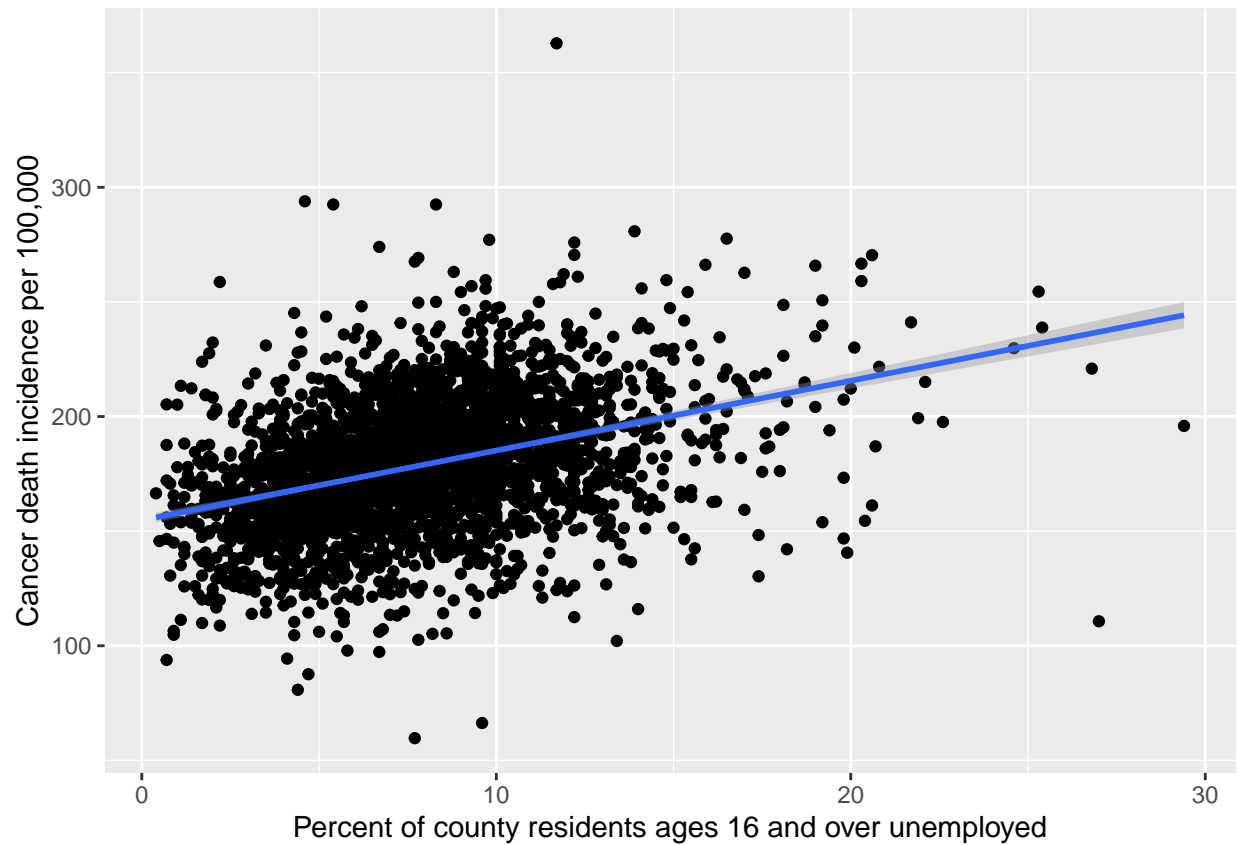
```
ggplot(data = cancer, aes(x = pctpubliccoverage, y = target_deathrate))+
  geom_point()+
  geom_smooth(method = lm)+
  labs(x="Percent of county residents with government-provided health insurance", y= "Cancer death inci

## `geom_smooth()` using formula 'y ~ x'
```



```
ggplot(data = cancer, aes(x = pctunemployed16_over, y = target_deathrate))+
  geom_point()+
  geom_smooth(method = lm)+
  labs(x="Percent of county residents ages 16 and over unemployed", y= "Cancer death incidence per 100,000")

## `geom_smooth()` using formula 'y ~ x'
```

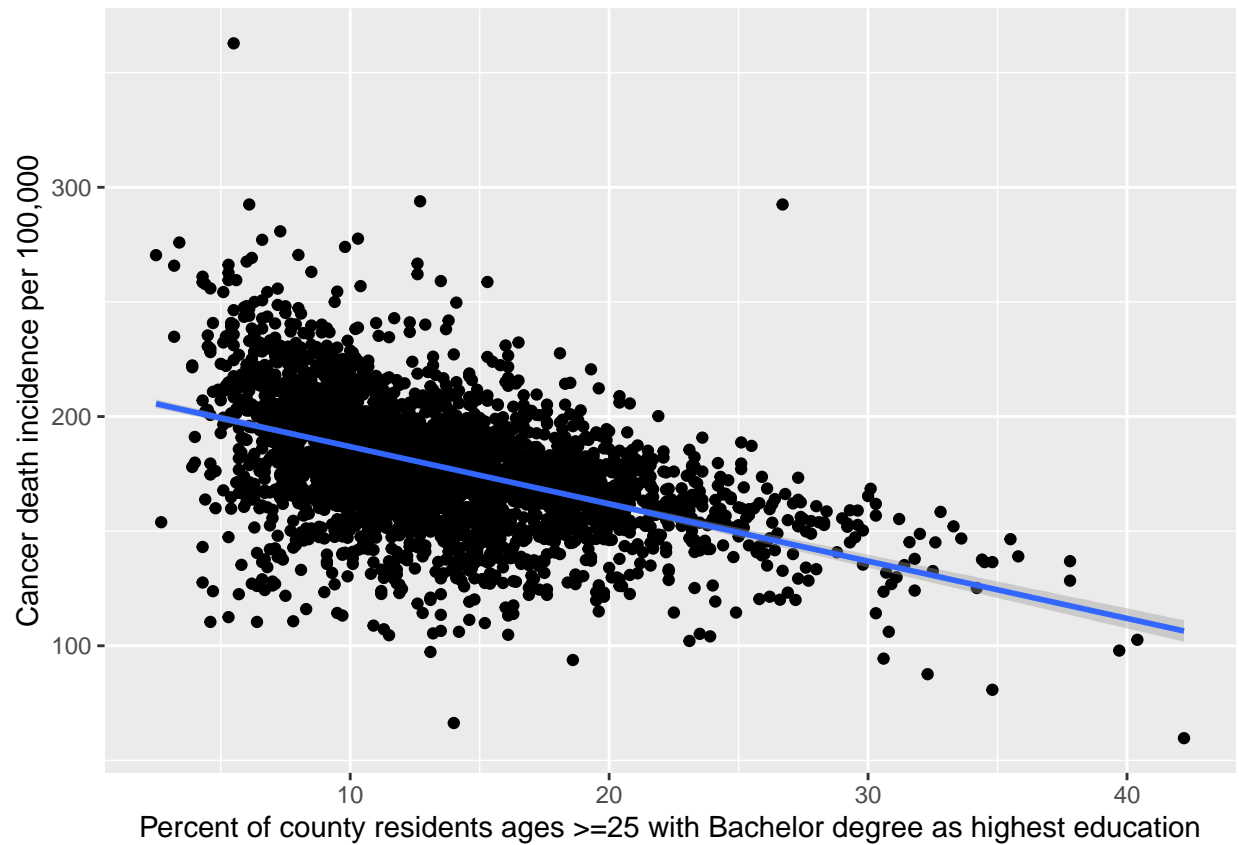



Plot negative correlations:

pctbachdeg25_over -.44 medincome -0.38 pctemployed16_over -0.37 pctprivatecoverage -0.34
pctprivatecoveragealone -0.32

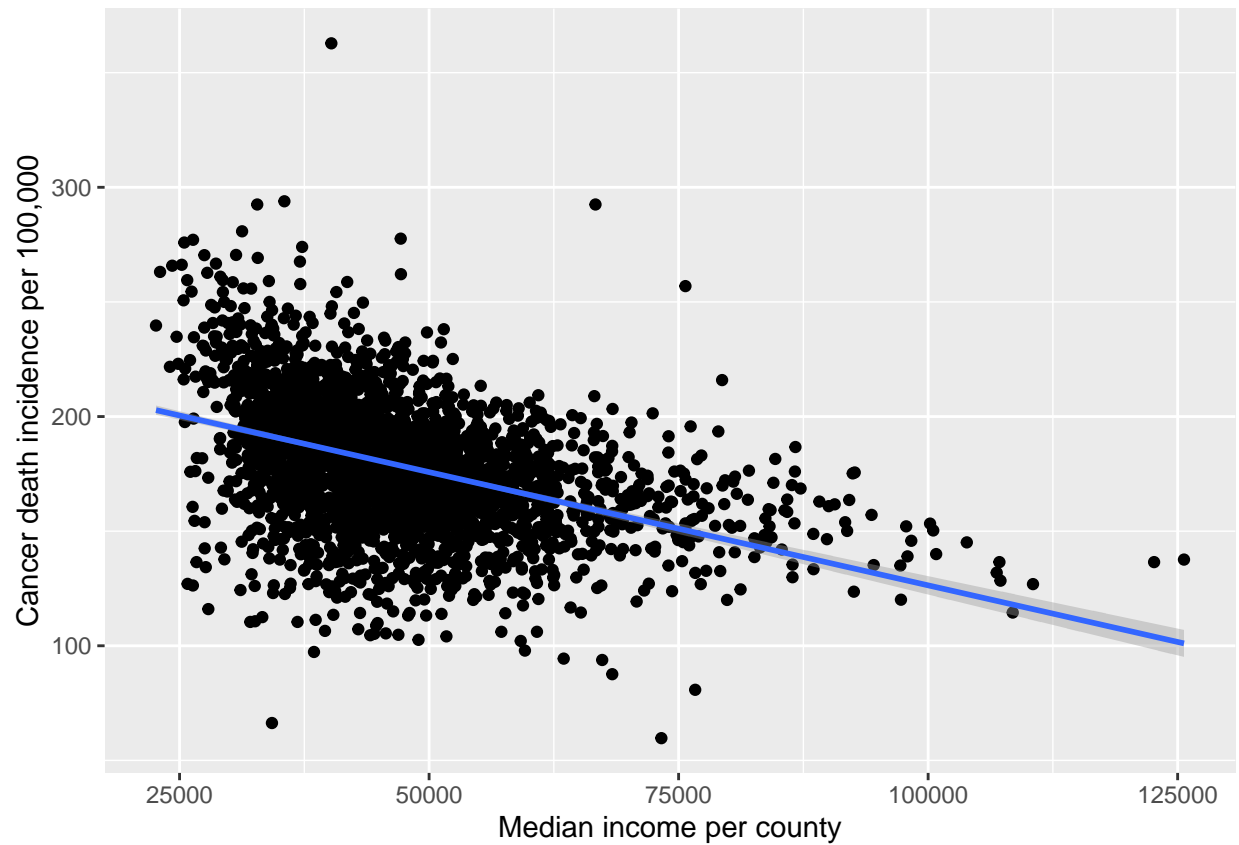
```
ggplot(data = cancer, aes(x = pctbachdeg25_over, y = target_deathrate))+
  geom_point()+
  geom_smooth(method = lm)+
  labs(x="Percent of county residents ages >=25 with Bachelor degree as highest education", y= "Cancer death incidence per 100,000")

## `geom_smooth()` using formula 'y ~ x'
```



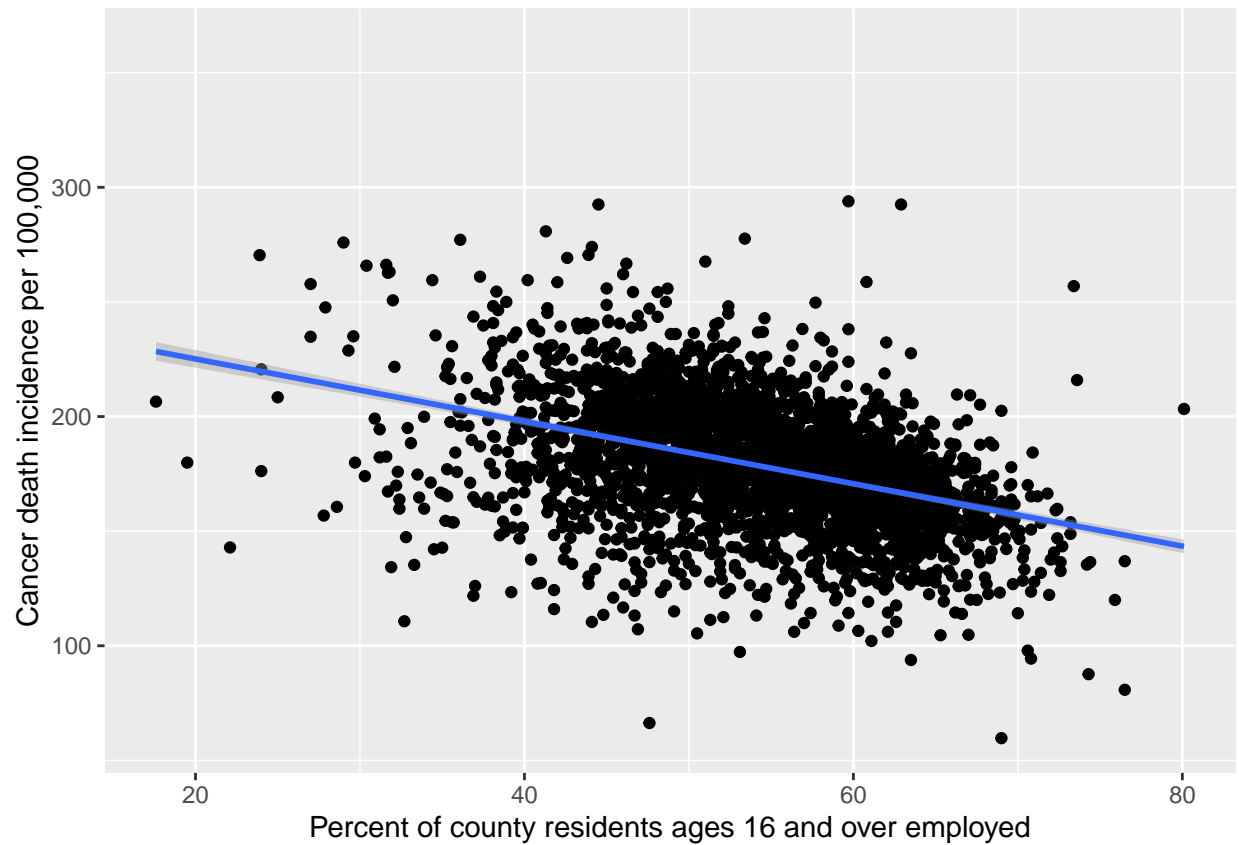
```
ggplot(data = cancer, aes(x = medincome, y = target_deathrate))+
  geom_point()+
  geom_smooth(method = lm)+
  labs(x="Median income per county", y= "Cancer death incidence per 100,000")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



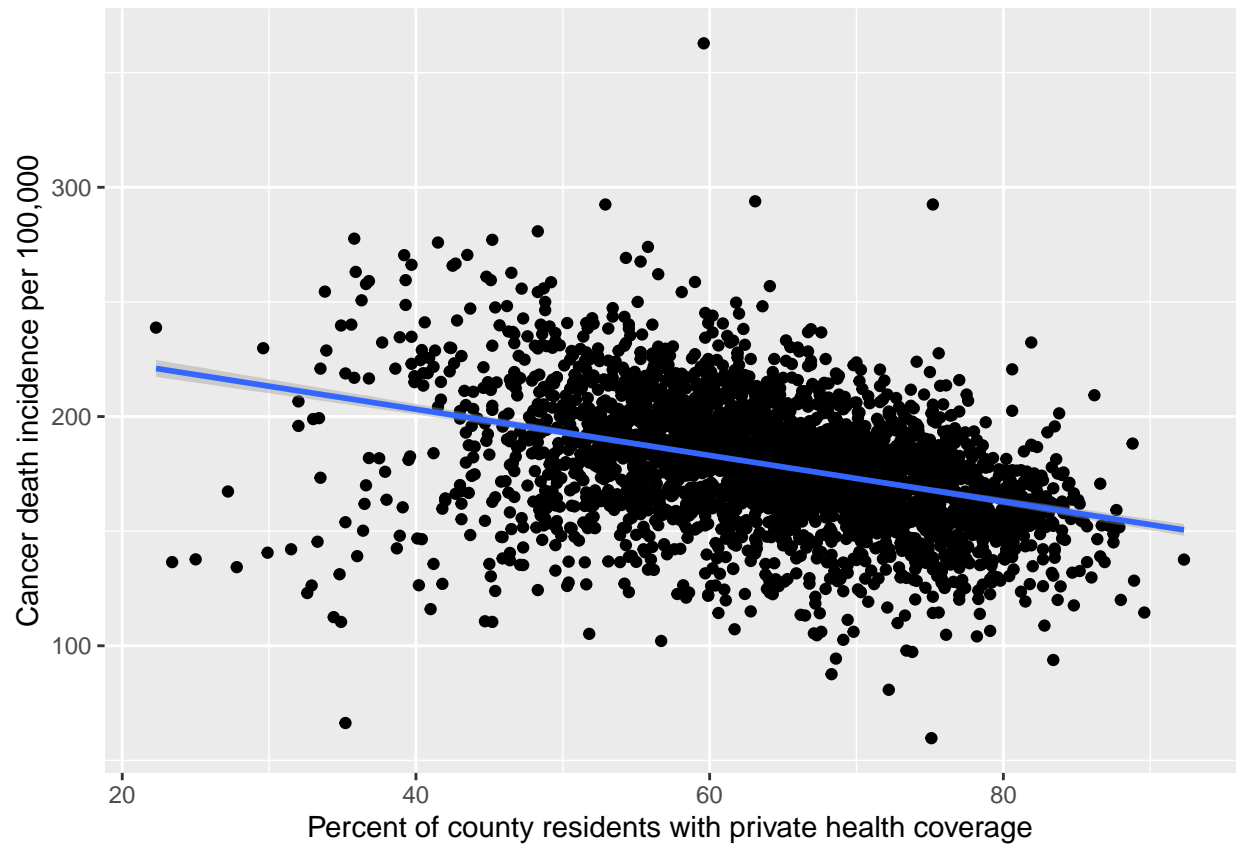
```
ggplot(data = cancer, aes(x = pctemployed16_over, y = target_deathrate))+
  geom_point()+
  geom_smooth(method = lm)+
  labs(x="Percent of county residents ages 16 and over employed", y= "Cancer death incidence per 100,000")

## `geom_smooth()` using formula 'y ~ x'
## Warning: Removed 152 rows containing non-finite values (stat_smooth).
## Warning: Removed 152 rows containing missing values (geom_point).
```



```
ggplot(data = cancer, aes(x = pctprivatecoverage, y = target_deathrate))+
  geom_point()+
  geom_smooth(method = lm)+
  labs(x="Percent of county residents with private health coverage", y= "Cancer death incidence per 100,000")

## `geom_smooth()` using formula 'y ~ x'
```

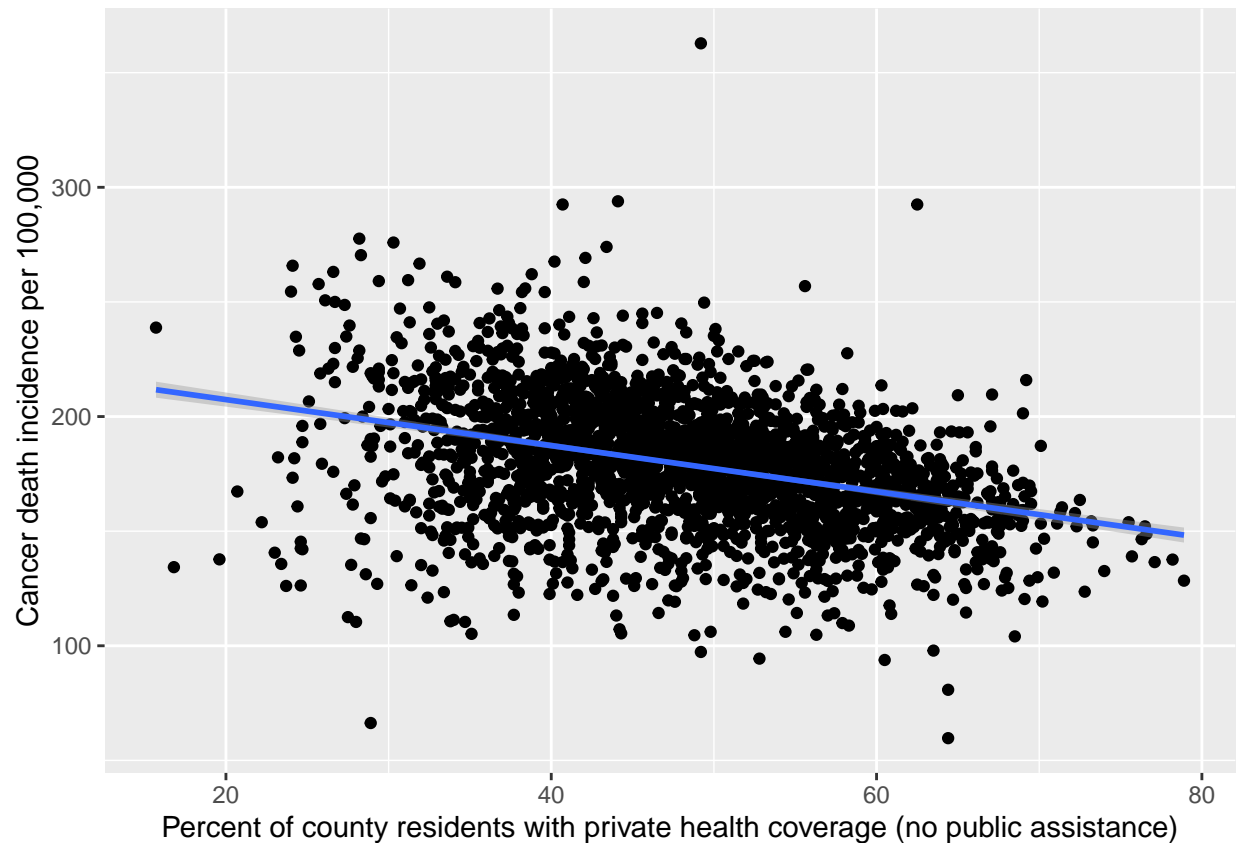


```
ggplot(data = cancer, aes(x = pctprivatecoveragealone, y = target_deathrate))+
  geom_point()+
  geom_smooth(method = lm)+
  labs(x="Percent of county residents with private health coverage (no public assistance)", y= "Cancer o
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 609 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 609 rows containing missing values (geom_point).
```



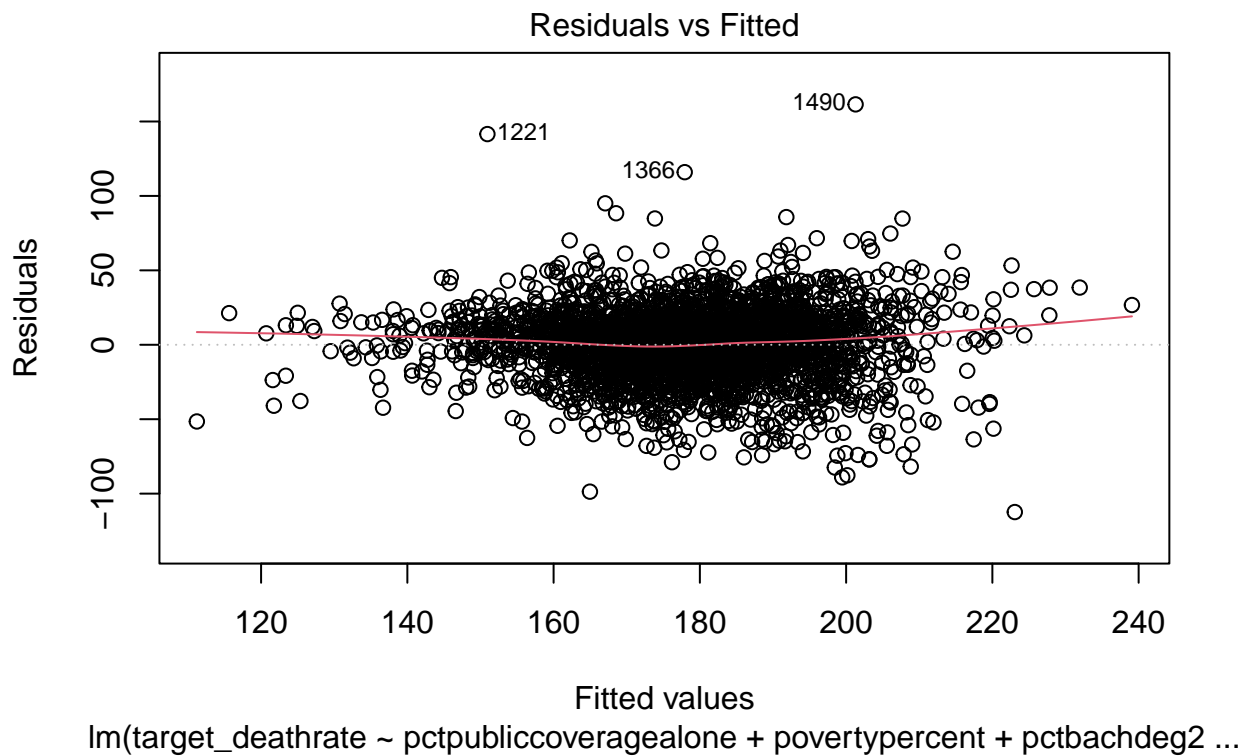
```
canc_fit_01 <- cancer %>% lm(target_deathrate ~ pctpubliccoveragealone + povertypercent + pctbachdeg25_
canc_fit_01_se <- cancf_fit_01 %>% vcovHC(type = "HC1") %>% diag() %>% sqrt()
stargazer(canc_fit_01, type="text", se = list(canc_fit_01_se))
```

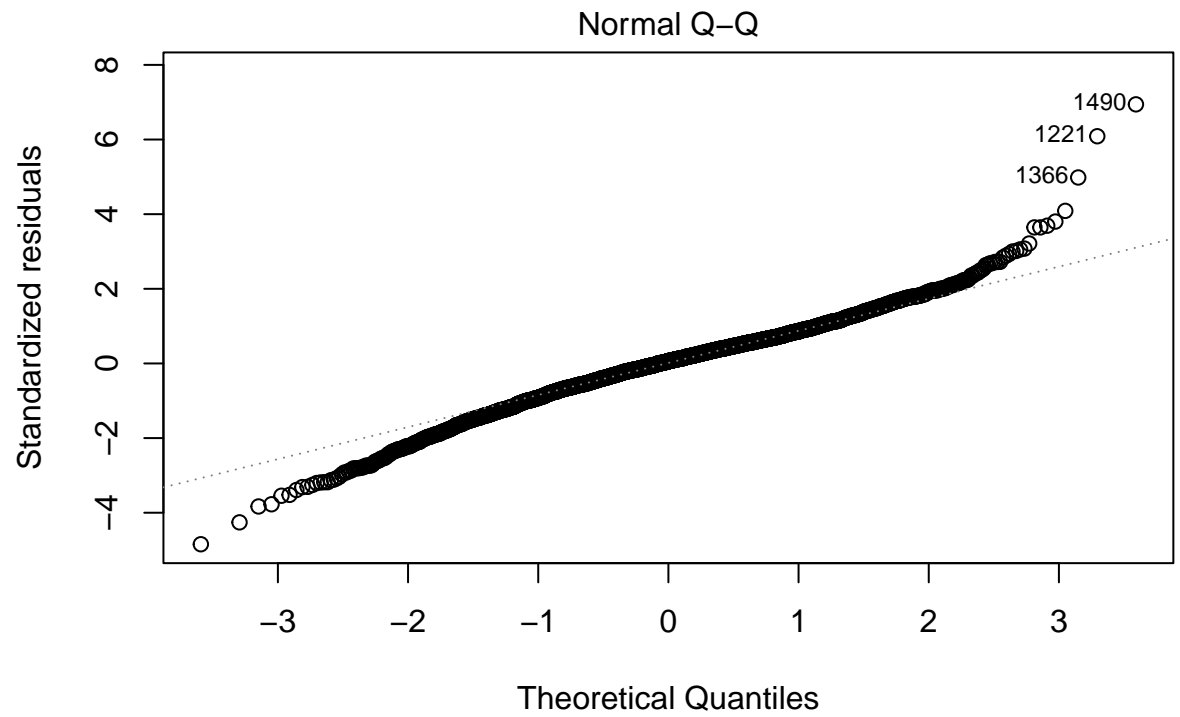
```
##
## =====
##                               Dependent variable:
##                               -----
##                               target_deathrate
##                               -----
## pctpubliccoveragealone        1.386***
##                               (0.189)
##
## povertypercent                1.047***
##                               (0.134)
##
## pctbachdeg25_over             -1.798***
##                               (0.104)
##
## pctprivatecoverage            0.767***
##                               (0.117)
##
## Constant                      108.875***
##                               (10.990)
##
## -----
```

```
## Observations          3,047
## R2                    0.297
## Adjusted R2           0.296
## Residual Std. Error   23.282 (df = 3042)
## F Statistic           321.425*** (df = 4; 3042)
## =====
## Note:                  *p<0.1; **p<0.05; ***p<0.01
coeftest(canc_fit_01, vconv = vcovHC(type = "HC1"))

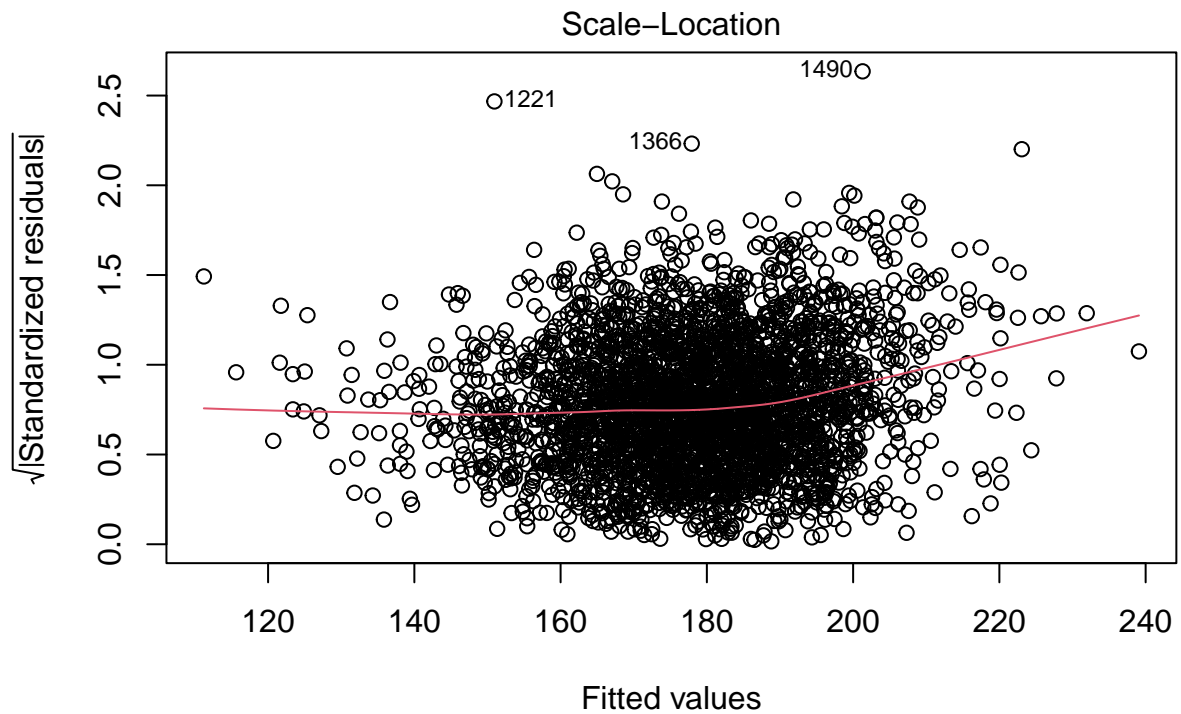
##
## t test of coefficients:
##
##               Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)    108.874598   9.028312  12.0592 < 2.2e-16 ***
## pctpubliccoveragealone  1.385756   0.157001   8.8264 < 2.2e-16 ***
## povertypercent     1.046783   0.120059   8.7189 < 2.2e-16 ***
## pctbachdeg25_over   -1.797838   0.099956  -17.9863 < 2.2e-16 ***
## pctprivatecoverage   0.766663   0.094974   8.0724 9.829e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

canc_fit_01_vif = ols_vif_tol(canc_fit_01)
plot(canc_fit_01)
```

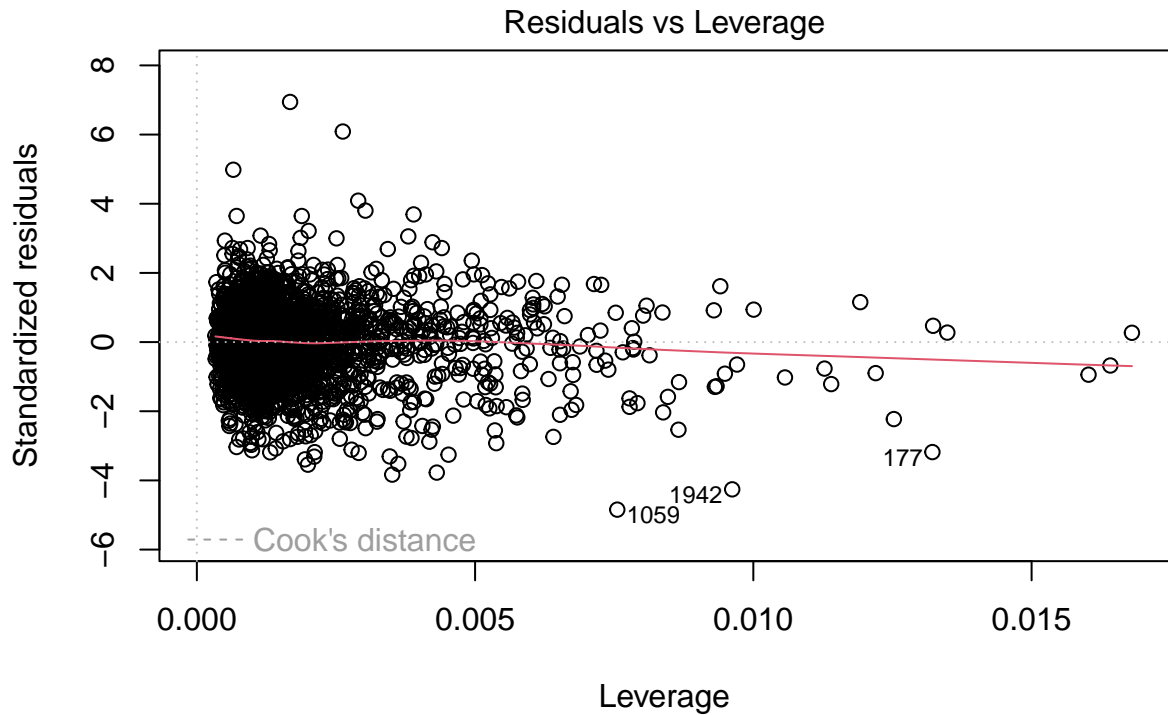




lm(target_deathrate ~ pctpubliccoveragealone + povertypercent + pctbachdeg2 ...



lm(target_deathrate ~ pctpubliccoveragealone + povertypercent + pctbachdeg2 ...



lm(target_deathrate ~ pctpubliccoveragealone + povertypercent + pctbachdeg2 ...

```
lmtest::bptest(canc_fit_01)
```

```
##
## studentized Breusch-Pagan test
##
## data:  canc_fit_01
## BP = 119.38, df = 4, p-value < 2.2e-16
```