

---

## Control confounding with regression methods

In this chapter we cover how to use regression models to control for confounding for a binary outcome. We cover the following models:

- Unconditional logistic regression
- Conditional logistic regression
- Poisson regression
- Cox proportional hazards regression model

### 6.0.3 Interpreting the regression coefficients

To interpret the regression coefficient, you must understand the transformation of  $y$ . Consider a linear regression model where  $y$  is a continuous outcome and the predictor  $x$  is a dichotomous variable. For an unexposed subject  $x = 0$  and for an exposed subject  $x = 1$ .

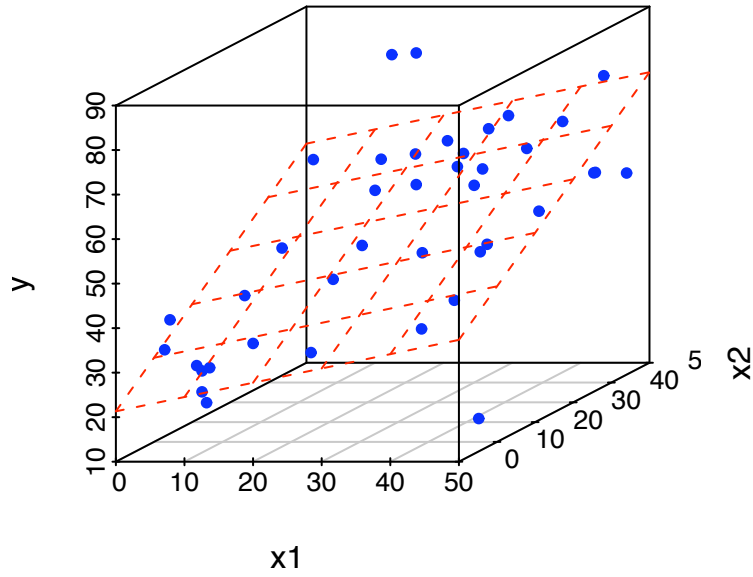
$$y = a_0 + a_1x$$

To interpret the coefficient  $a_1$  take the difference of the equation when  $x = 1$  and the equation when  $x = 0$ .

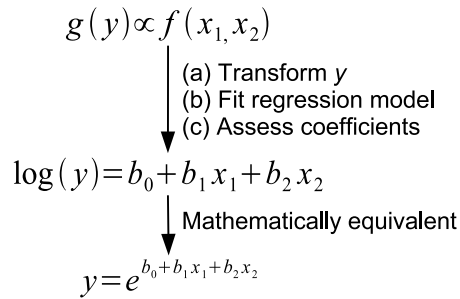
$$\begin{aligned}y_e &= a_0 + a_1(1) \\ -[y_u &= a_0 + a_1(0)] \\ (y_e - y_u) &= (a_0 - a_0) + (a_1 - 0) \\ y_e - y_u &= a_1\end{aligned}$$

Therefore, for this linear regression equation, the coefficient  $a_1$  is equal to the change in  $y$  ( $y_e - y_u$ ) for a unit change in  $x$  ( $1 - 0$ ).

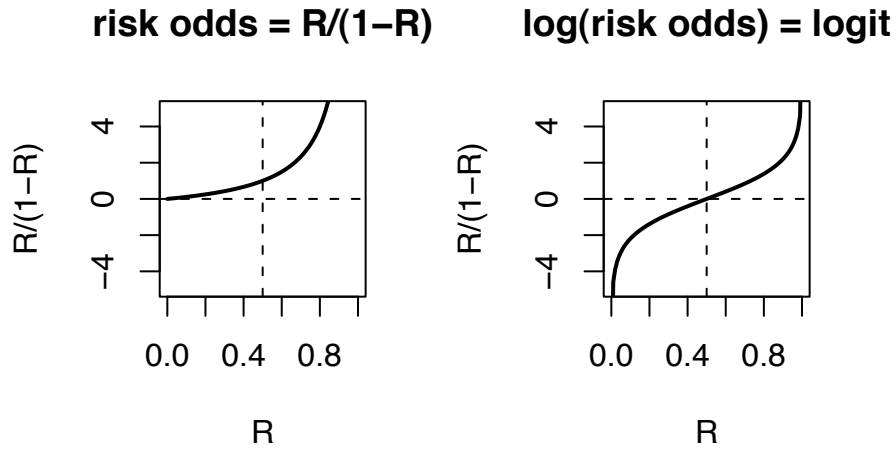
Now consider the same data but we take the natural logarithm of  $y$  first, and repeat the same analysis.



**Fig. 6.1.** A general linear model ( $y = a_0 + a_1x_1 + a_2x_2$ ) is fitted to a cloud of data points  $(x_1, x_2, y)$ . The model coefficients  $(a_0, a_1, a_2)$  are the intercept and slopes of a fitted plane that minimize the distance to the observed data



**Fig. 6.2.** In epidemiology, the dependent variable ( $y$ ) is usually transformed (for example, natural logarithm), and a regression model is fit.



**Fig. 6.3.** The logit transformation is a double transformation. First, the odds transformation ( $R/(1-R)$ ) unbounds the probabilities near 1; second, the logit transformation ( $\log(\text{odds})$ ) additionally unbounds the probabilities near 0.

$$\begin{aligned} \log(y_e) &= a_0 + a_1(1) \\ -[\log(y_u) &= a_0 + a_1(0)] \\ (\log(y_e) - \log(y_u)) &= (a_0 - a_0) + (a_1 - 0) \\ \log\left(\frac{y_e}{y_u}\right) &= a_1 \\ \frac{y_e}{y_u} &= e^{a_1} \end{aligned}$$

Therefore, now the coefficient  $a_1$  is equal to the change in the  $\log(y)$  ( $\log(y_e) - \log(y_u)$ ) for a unit change in  $x$  ( $1 - 0$ ). Equivalently,  $a_1$  is equal to the  $\log(y_1/y_0)$ , or  $e^{a_1} = y_1/y_0$ .

## 6.1 Linear regression

## 6.2 Logistic regression

$$\begin{aligned} \log(R_1/(1 - R_1)) &= a_0 + a_1(1) \\ -[\log(R_0/(1 - R_0)) &= a_0 + a_1(0)] \\ (\log[R_1/(1 - R_1)] - \log[R_0/(1 - R_0)]) &= (a_0 - a_0) + (a_1 - 0) \\ \log\left(\frac{R_1/(1 - R_1)}{R_0/(1 - R_0)}\right) &= a_1 \\ \frac{R_1/(1 - R_1)}{R_0/(1 - R_0)} &= e^{a_1} \end{aligned}$$

### 6.2.1 Unconditional logistic regression

```

or.glm <- function(x, conf.level = 0.95){
  sumx <- summary(x)$coefficients
  Z <- qnorm((1 + conf.level)/2)
  expor <- exp(sumx[,"Estimate"])
  lcl <- exp(sumx[,"Estimate"] - Z*sumx[,"Std. Error"])
  ucl <- exp(sumx[,"Estimate"] + Z*sumx[,"Std. Error"])
  coef <- sumx
  or <- cbind(coef=sumx[,"Estimate"], "exp(coef)"=expor, lcl, ucl)
  list(coef = coef, or = or)
}

> edat <- read.table("http://www.medepi.net/data/evans.txt",
+                   header = TRUE, sep = "")
> ##prepare nominal categorical variables
> edat$chd <- factor(edat$chd, levels=0:1, labels=c("No","Yes"))
> edat$cat <- factor(edat$cat, levels=0:1, labels=c("Normal","High"))
> edat$smk <- factor(edat$smk, levels=0:1, labels=c("Never","Ever"))
> edat$ecg <- factor(edat$ecg, levels=0:1, labels=c("Normal","Abnormal"))
> edat$hpt <- factor(edat$hpt, levels=0:1, labels=c("No","Yes"))
> str(edat)
'data.frame': 609 obs. of 12 variables:
 $ id : int  21 31 51 71 74 91 111 131 141 191 ...
 $ chd: Factor w/ 2 levels "No","Yes": 1 1 2 1 1 1 2 1 1 1 ...
 $ cat: Factor w/ 2 levels "Normal","High": 1 1 2 2 1 1 1 1 1 1 ...
 $ age: int  56 43 56 64 49 46 52 63 42 55 ...
 $ chl: int  270 159 201 179 243 252 179 217 176 250 ...
 $ smk: Factor w/ 2 levels "Never","Ever": 1 2 2 2 2 2 2 1 2 1 ...
 $ ecg: Factor w/ 2 levels "Normal","Abnormal": 1 1 2 1 1 1 2 1 1 2 ...
 $ dbp: int  80 74 112 100 82 88 80 92 76 114 ...
 $ sbp: int  138 128 164 200 145 142 128 135 114 182 ...
 $ hpt: Factor w/ 2 levels "No","Yes": 1 1 2 2 1 1 1 1 1 2 ...
 $ ch  : int  0 0 1 1 0 0 0 0 0 0 ...
 $ cc  : int  0 0 201 179 0 0 0 0 0 0 ...
> m1 <- glm(chd ~ cat, family = binomial, data = edat)

```

```

> or.glm(m1)
$coef
              Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -2.309380  0.1580652 -14.610301 2.414484e-48
catHigh      1.051340  0.2693473   3.903288 9.489454e-05

$or
              coef exp(coef)      lcl      ucl
(Intercept) -2.309380 0.0993228 0.07286242 0.1353924
catHigh      1.051340 2.8614833 1.68780545 4.8513212

> m2 <- glm(chd ~ cut(age,2), family = binomial, data = edat)
> or.glm(m2)
$coef
              Estimate Std. Error   z value
(Intercept)   -2.3253998  0.1677715 -13.860514
cut(age, 2)(58,76]  0.8566617  0.2580534   3.319708
              Pr(>|z|)
(Intercept)    1.09884e-43
cut(age, 2)(58,76] 9.01117e-04

$or
              coef exp(coef)      lcl      ucl
(Intercept)   -2.3253998 0.09774436 0.07035328 0.1357998
cut(age, 2)(58,76]  0.8566617 2.35528500 1.42032606 3.9057000

> m3 <- glm(chd ~ cut(chl,c(90,200,360)) , family = binomial, data = edat)
> or.glm(m3)
$coef
              Estimate Std. Error
(Intercept)   -2.4723279  0.2328505
cut(chl, c(90, 200, 360))(200,360]  0.6970432  0.2777579
              z value   Pr(>|z|)
(Intercept)   -10.617662 2.466628e-26
cut(chl, c(90, 200, 360))(200,360]  2.509535 1.208902e-02

$or
              coef exp(coef)
(Intercept)   -2.4723279 0.08438819
cut(chl, c(90, 200, 360))(200,360]  0.6970432 2.00780731
              lcl      ucl
(Intercept)   0.05346615 0.1331939
cut(chl, c(90, 200, 360))(200,360] 1.16491453 3.4605888

> m4 <- glm(chd ~ smk, family = binomial, data = edat)

```

```

> or.glm(m4)
$coef
              Estimate Std. Error  z value    Pr(>|z|)
(Intercept) -2.4897966  0.2523916 -9.864817 5.914218e-23
smkEver      0.6706382  0.2919298  2.297258 2.160405e-02

$or
              coef exp(coef)      lcl      ucl
(Intercept) -2.4897966 0.08292683 0.05056604 0.1359976
smkEver      0.6706382 1.95548490 1.10347715 3.4653379

> m5 <- glm(chd ~ hpt, family = binomial, data = edat)
> or.glm(m5)
$coef
              Estimate Std. Error  z value    Pr(>|z|)
(Intercept) -2.4546929  0.1969308 -12.464748 1.162384e-35
hptYes       0.8593067  0.2583690  3.325889 8.813687e-04

$or
              coef exp(coef)      lcl      ucl
(Intercept) -2.4546929 0.08588957 0.05838652 0.1263480
hptYes       0.8593067 2.36152291 1.42320702 3.9184675

```

### 6.2.2 Conditional logistic regression

```

> mdat <- read.table("http://www.medepi.net/data/mi.txt", sep="")
> names(mdat) <- c("match", "person", "mi", "smk", "sbp", "ecg")
> mdat$mi <- factor(mdat$mi, levels=0:1, labels=c("No", "Yes"))
> mdat$smk <- factor(mdat$smk, levels=0:1, labels=c("Not current", "Current"))
> mdat$ecg <- factor(mdat$ecg, levels=0:1, labels=c("Normal", "Abnormal"))
> str(mdat)
'data.frame': 117 obs. of 6 variables:
 $ match : int  1 1 1 2 2 2 3 3 3 4 ...
 $ person: int  1 2 3 4 5 6 7 8 9 10 ...
 $ mi    : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 1 2 1 1 2 ...
 $ smk   : Factor w/ 2 levels "Not current",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ sbp   : int  160 140 120 160 140 120 160 140 120 160 ...
 $ ecg   : Factor w/ 2 levels "Normal","Abnormal": 2 1 1 2 1 1 1 1 1 1 ...
>
> library(survival)
> clog1 <- clogit(unclass(mi)~smk+strata(match), data=mdat)
> summary(clog1)
Call:
coxph(formula = Surv(rep(1, 117L), unclass(mi)) ~ smk + strata(match),
      data = mdat, method = "exact")

```

```

n= 117
      coef exp(coef) se(coef)    z    p
smkCurrent 0.843      2.32   0.466 1.81 0.07

      exp(coef) exp(-coef) lower .95 upper .95
smkCurrent      2.32      0.43   0.932   5.8

Rsquare= 0.028 (max possible= 0.519 )
Likelihood ratio test= 3.37 on 1 df, p=0.0665
Wald test          = 3.27 on 1 df, p=0.0704
Score (logrank) test = 3.43 on 1 df, p=0.0641

> clog2 <- clogit(unclass(mi)~cut(sbp,2)+strata(match),data=mdat)
> summary(clog2)
Call:
coxph(formula = Surv(rep(1, 117L), unclass(mi)) ~ cut(sbp, 2) +
      strata(match), data = mdat, method = "exact")

n= 117
      coef exp(coef) se(coef)    z    p
cut(sbp, 2)(140,160] 2.04      7.67   0.458 4.44 8.9e-06

      exp(coef) exp(-coef) lower .95 upper .95
cut(sbp, 2)(140,160]      7.67      0.130   3.12   18.8

Rsquare= 0.198 (max possible= 0.519 )
Likelihood ratio test= 25.8 on 1 df, p=3.72e-07
Wald test          = 19.7 on 1 df, p=8.86e-06
Score (logrank) test = 27.6 on 1 df, p=1.50e-07

> clog3 <- clogit(unclass(mi)~ecg+strata(match),data=mdat)
> summary(clog3)
Call:
coxph(formula = Surv(rep(1, 117L), unclass(mi)) ~ ecg + strata(match),
      data = mdat, method = "exact")

n= 117
      coef exp(coef) se(coef)    z    p
ecgAbnormal 2.06      7.88   0.785 2.63 0.0086

      exp(coef) exp(-coef) lower .95 upper .95
ecgAbnormal      7.88      0.127   1.69   36.8

Rsquare= 0.078 (max possible= 0.519 )

```

Likelihood ratio test= 9.51 on 1 df, p=0.00204  
 Wald test = 6.91 on 1 df, p=0.00857  
 Score (logrank) test = 9.37 on 1 df, p=0.0022

```
> clog4 <- clogit(unclass(mi)~smk+sbp+ecg+strata(match),data=mdat)
> summary(clog4)
```

Call:

```
coxph(formula = Surv(rep(1, 117L), unclass(mi)) ~ smk + sbp +
      ecg + strata(match), data = mdat, method = "exact")
```

n= 117

	coef	exp(coef)	se(coef)	z	p
smkCurrent	0.7291	2.07	0.5613	1.30	0.1900
sbp	0.0456	1.05	0.0152	2.99	0.0028
ecgAbnormal	1.5993	4.95	0.8534	1.87	0.0610

	exp(coef)	exp(-coef)	lower .95	upper .95
smkCurrent	2.07	0.482	0.69	6.23
sbp	1.05	0.955	1.02	1.08
ecgAbnormal	4.95	0.202	0.93	26.36

Rsquare= 0.173 (max possible= 0.519 )

Likelihood ratio test= 22.2 on 3 df, p=5.92e-05

Wald test = 13.7 on 3 df, p=0.00338

Score (logrank) test = 19.7 on 3 df, p=0.000198

```
> clog5 <- clogit(unclass(mi)~smk+cut(sbp,2)+ecg+strata(match),data=mdat)
> summary(clog5)
```

Call:

```
coxph(formula = Surv(rep(1, 117L), unclass(mi)) ~ smk + cut(sbp,
      2) + ecg + strata(match), data = mdat, method = "exact")
```

n= 117

	coef	exp(coef)	se(coef)	z	p
smkCurrent	0.645	1.91	0.609	1.06	0.29000
cut(sbp, 2)(140,160]	1.979	7.23	0.502	3.94	0.00008
ecgAbnormal	1.947	7.01	0.992	1.96	0.05000

	exp(coef)	exp(-coef)	lower .95	upper .95
smkCurrent	1.91	0.525	0.578	6.29
cut(sbp, 2)(140,160]	7.23	0.138	2.706	19.34
ecgAbnormal	7.01	0.143	1.003	48.94

Rsquare= 0.237 (max possible= 0.519 )

Likelihood ratio test= 31.7 on 3 df, p=5.99e-07

Wald test = 17.4 on 3 df, p=0.00058  
 Score (logrank) test = 31 on 3 df, p=8.61e-07

### 6.3 Poisson regression

$$\log(\text{rate}) = a_0 + a_1x$$

$$\log(r_1) = a_0 + a_1(1)$$

$$-[\log(r_0) = a_0 + a_1(0)]$$

$$\log(r_1) - \log(r_0) = (a_0 - a_0) + (a_1 - 0)$$

$$\log\left(\frac{r_1}{r_0}\right) = a_1$$

$$\frac{r_1}{r_0} = e^{a_1}$$

For fitting the Poisson regression model, the following relationship is important:

$$\log\left(\frac{\text{count}}{\text{person-time}}\right) = a_0 + a_1x$$

$$\log(\text{count}) - \log(\text{person-time}) = a_0 + a_1x$$

$$\log(\text{count}) = a_0 + a_1x + \log(\text{person-time})$$

where  $\log(\text{person-years})$  is called the *offset*. The offset is considered a fixed quantity and is not fit in the statistical model; however, we will need to specify this in the Poisson model in R. We'll see this in the examples that follow.

### 6.4 Indirect standardization using Poisson regression

The direct standardization of rates involves using weights from a standard population. In contrast, the indirect standardization of rates involves using the crude and age-specific rates from a standard population. The calculation involves two components:

1. Calculate the standardized incidence ratio (SIR) (also called standardized mortality ratio [SMR]).
2. Multiply the SIR and the standard crude rate

$$\begin{aligned}
 ISR &= (SIR)(R_{\text{crude}}^S) \\
 &= \left( \frac{\text{Observed}}{\text{Expected}} \right) R_{\text{crude}}^S \\
 &= \left( \frac{\sum_i A_i}{\sum_i PT_i R_i^S} \right) R_{\text{crude}}^S
 \end{aligned}$$

In this example, we use the dataset of U.S. white male population estimates and male cancer deaths in 1940 compared to 1960.

```

> #enter data
> dth60 <- c(141, 926, 1253, 1080, 1869, 4891, 14956, 30888,
+           41725, 26501, 5928)
> pop60 <- c(1784033, 7065148, 15658730, 10482916, 9939972,
+           10563872, 9114202, 6850263, 4702482, 1874619,
+           330915)
> dth40 <- c(45, 201, 320, 670, 1126, 3160, 9723, 17935,
+           22179, 13461, 2238)
> pop40 <- c(906897, 3794573, 10003544, 10629526, 9465330,
+           8249558, 7294330, 5022499, 2920220, 1019504,
+           142532)
> #housekeeping
> tab <- cbind(pop40, dth40, pop60, dth60)
> agelabs <- c("<1", "1-4", "5-14", "15-24", "25-34",
+           "35-44", "45-54", "55-64", "65-74", "75-84",
+           "85+")
> rownames(tab) <- agelabs
> #calculate crude rates
> CDR.1940 <- sum(dth40)/sum(pop40)
> CDR.1960 <- sum(dth60)/sum(pop60)
> #display data and crude rates
> tab
      pop40 dth40  pop60 dth60
<1      906897   45 1784033   141
1-4     3794573  201 7065148   926
5-14   10003544  320 15658730  1253
15-24  10629526  670 10482916  1080
25-34  9465330  1126 9939972  1869
35-44  8249558  3160 10563872  4891
45-54  7294330  9723 9114202 14956
55-64  5022499 17935 6850263 30888
65-74  2920220 22179 4702482 41725
75-84  1019504 13461 1874619 26501
85+    142532  2238  330915  5928
> round(100000*c(CDR.1940=CDR.1940, CDR.1960=CDR.1960), 1)
CDR.1940 CDR.1960

```

119.5 166.1

Now, we calculate the indirect standardized rate for 1940 using the 1960 data as the standard. Using the data we prepared above, we do the calculations in two steps:

```
> # sir calculation
> Ri.1960 <- dth60/pop60
> CDR.1960 <- sum(dth60)/sum(pop60)
> SIR.1940 <- sum(dth40)/sum(pop40*Ri.1960)
> ISR.1940 <- SIR.1940*CDR.1960
> c(SIR.1940=SIR.1940, ISR.1940=100000*ISR.1940,
+   CDR.1960=100000*CDR.1960)
      SIR.1940   ISR.1940   CDR.1960
0.8305351 137.9414537 166.0874444
```

Now, to use Poisson regression we need to prepare our male cancer death data as a data frame, and set 1960 as the standard (reference) population.

```
> #create data frame for poisson regression
> year <- factor(c(rep(1960, 11), rep(1940, 11)),
+               levels = c(1960, 1940)) #1960 as reference
> pop <- c(pop60, pop40)
> deaths <- c(dth60, dth40)
> age <- factor(rep(agemlabs, 2), levels=agemlabs)
> cad <- data.frame(year, pop, deaths, age)
> cad
   year   pop deaths age
1  1960 1784033   141  <1
2  1960  7065148   926  1-4
3  1960 15658730  1253  5-14
...
20 1940  2920220 22179 65-74
21 1940  1019504 13461 75-84
22 1940   142532  2238  85+
```

We are now in the position to use the Poisson regression model to estimate the crude death rates for the 1940 and 1960 data.

```
> pmod <- glm(deaths~year, family = poisson(link="log"), data = cad,
+             offset = log(pop))
> summary(pmod)
```

Call:

```
glm(formula = deaths ~ year, family = poisson(link = "log"),
     data = cad, offset = log(pop))
```

Deviance Residuals:

	Min	1Q	Median	3Q	Max
	-204.71	-128.34	-55.75	130.06	268.34

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-6.400411	0.002772	-2309.11	<2e-16 ***
year1940	-0.328958	0.004664	-70.53	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 487587 on 21 degrees of freedom  
 Residual deviance: 482471 on 20 degrees of freedom  
 AIC: 482690

Number of Fisher Scoring iterations: 6

```
> pmod$coef
(Intercept)   year1940
-6.4004110  -0.3289584
> cdr.1940 <- exp(pmod$coeff[1] + pmod$coeff[2])
> cdr.1940*100000 #from model
(Intercept)
 119.5286
> CDR.1940*100000 #from arithmetic
[1] 119.5286
> cdr.1960 <- exp(pmod$coeff[1])
> cdr.1960*100000 #from model
(Intercept)
 166.0874
> CDR.1960*100000 #from arithmetic
[1] 166.0874
```

We now use the Poisson model to calculate an adjusted rate ratio comparing the 1940 population to the 1960 population (standard). This rate ratio closely approximates the SIR that, when multiplied by the 1960 crude rate, gives us the indirect standardized rate.

```
> pmod2 <- glm(deaths ~ year + age, family= poisson(link="log"),
+             data = cad, offset = log(pop))
> summary(pmod2)
```

Call:

```
glm(formula = deaths ~ year + age, family = poisson(link = "log"),
    data = cad, offset = log(pop))
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-10.51100	-3.69907	-0.02425	3.15316	8.81875

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-9.521870	0.073337	-129.838	< 2e-16 ***
year1940	-0.182227	0.004679	-38.945	< 2e-16 ***
age1-4	0.408585	0.079143	5.163	2.44e-07 ***
age5-14	-0.110780	0.077538	-1.429	0.15308
age15-24	0.211467	0.077125	2.742	0.00611 **
age25-34	0.830261	0.075569	10.987	< 2e-16 ***
age35-44	1.841193	0.074167	24.825	< 2e-16 ***
age45-54	3.099207	0.073601	42.108	< 2e-16 ***
age55-64	4.101147	0.073464	55.825	< 2e-16 ***
age65-74	4.806312	0.073430	65.454	< 2e-16 ***
age75-84	5.299837	0.073494	72.112	< 2e-16 ***
age85+	5.513262	0.074154	74.349	< 2e-16 ***

---

Signif. codes: 0 \*\*\* 0.001 \*\* 0.01 \* 0.05 . 0.1 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 487587.25 on 21 degrees of freedom  
 Residual deviance: 517.29 on 10 degrees of freedom  
 AIC: 755.8

Number of Fisher Scoring iterations: 4

```
> pmod2$coef
(Intercept)  year1940    age1-4    age5-14    age15-24
-9.5218697 -0.1822271  0.4085846 -0.1107800  0.2114672
  age25-34  age35-44  age45-54  age55-64  age65-74
  0.8302607  1.8411928  3.0992069  4.1011465  4.8063119
  age75-84  age85+
  5.2998370  5.5132617
> exp(pmod2$coeff[2]) #SIR from Poisson model
year1940
0.833412
> SIR.1940      #SIR from arithmetic
[1] 0.8305351
```

Poisson regression seems like a lot of work. However, an advantage is that we have standard errors estimates for constructing confidence intervals.

```

> cdr.1960 #crude death rate for 1960 from Poisson model
(Intercept)
0.001660874
> conf.level <- 0.95
> Z <- qnorm((1+conf.level)/2)
> sy <- summary(pmod2)$coef #this includes standard errors
> log.sir.1940 <- sy["year1940", "Estimate"]
> sir.1940 <- exp(log.sir.1940)
> SE.log.sir.1940 <- sy["year1940", "Std. Error"]
> LCL.sir.1940 <- exp(log.sir.1940 - Z*SE.log.sir.1940)
> UCL.sir.1940 <- exp(log.sir.1940 + Z*SE.log.sir.1940)
> isr.1940 <- cdr.1960*c(sir.1940, LCL.sir.1940, UCL.sir.1940)
> names(isr.1940) <- c("ISR", "LCL", "UCL")
> round(isr.1940*100000, 1)
   ISR   LCL   UCL
138.4 137.2 139.7

```

Recall that the previously calculated  $ISR^{1940} = 137.9$  per 100,000 person-years.

## 6.5 Cox proportional hazards regression

```

library(survival)
pbc2 <- pbc
pbc2[pbc==9] <- NA
str(pbc2)

cmod1 <- coxph(Surv(time, status)~ascites, data=pbc2)
summary(cmod1)
plot(survfit(cmod1))

cmod2 <- coxph(Surv(time, status)~ascites+sex, data=pbc2)
summary(cmod2)
plot(survfit(cmod2))

sfit1 <- survfit(Surv(time, status)~ascites, data=pbc2)
plot(sfit1)

```