STATS 4250 Project

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

Data Set (colon)

Description

These are data from one of the first successful trials of adjuvant chemotherapy for colon cancer. Levamisole is a low-toxicity compound previously used to treat worm infestations in animals; 5-FU is a moderately toxic (as these things go) chemotherapy agent. There are two records per person, one for recurrence and one for death.

Variables

- id: id
- study: 1 for all patients
- rx: Treatment - Obs(ervation), Lev(amisole), Lev(amisole)+5-FU *sex: 1=male* age: in years
- obstruct: obstruction of colon by tumour
- perfor: perforation of colon
- adhere: adherence to nearby organs
- nodes: number of lymph nodes with detectable cancer
- time: days until event or censoring
- status: censoring status
- differ: differentiation of tumour (1=well, 2=moderate, 3=poor)
- extent: Extent of local spread (1=submucosa, 2=muscle, 3=serosa, 4=contiguous structures)
- surg: time from surgery to registration (0=short, 1=long)
- node4: more than 4 positive lymph nodes
- etype: event type: 1=recurrence,2=death

The event time is **time** (days until event or censoring), censor status is **status**. Here's what the data set looks like

```
library(survival)
library(survminer) # for ggcoxfunctional
```

```
## Loading required package: ggplot2
```

```
## Loading required package: ggpubr
```

```
## Loading required package: magrittr
```

```r
head(colon)
```

```
##   id study      rx sex age obstruct perfor adhere nodes status differ extent
## 1  1     1 Lev+5FU   1  43        0      0      0     5      1      2      3
## 2  1     1 Lev+5FU   1  43        0      0      0     5      1      2      3
## 3  2     1 Lev+5FU   1  63        0      0      0     1      0      2      3
## 4  2     1 Lev+5FU   1  63        0      0      0     1      0      2      3
## 5  3     1     Obs   0  71        0      0      1     7      1      2      2
## 6  3     1     Obs   0  71        0      0      1     7      1      2      2
##   surg node4 time etype
## 1    0     1 1521     2
## 2    0     1  968     1
## 3    0     0 3087     2
## 4    0     0 3087     1
## 5    0     1  963     2
## 6    0     1  542     1
```

Initial Thoughts

possible covariates:

- age
- sex
- rx
- obstruct
- perfor
- adhere
- nodes
- differ
- extent
- surg
- node4
- etype

```r
# this code will look at one-variable Cox PH models, and report
# the resulting beta coefficients, test statistics, and p-values
testCovariates <- c("age","sex","rx","obstruct","perfor","adhere","nodes","differ","extent","surg","node
univ_formulas <- sapply(testCovariates,function(x)as.formula(paste('Surv(time,status)~',x)))
univ_models <- lapply(univ_formulas,function(x){coxph(x,data=colon)})
# Extract data
univ_results <- lapply(univ_models,function(x){
  x<-summary(x)
  p.value<-signif(x$wald["pvalue"],digits=3)
  wald.test<-signif(x$wald["test"],digits=3)
  beta<-signif(x$coef[1], digits=3);#coeficient beta
  res<-c(beta, wald.test, p.value)
  names(res)<-c("beta", "wald.test","p.value")
  return(res)
#return(exp(cbind(coef(x),confint(x))))
  })
res <- t(as.data.frame(univ_results))
format(as.data.frame(res),scientific=FALSE)
```

```
##              beta wald.test                                                p.value
## age      -0.00244      0.76 0.382000000000000006217248937900876626372337341
## sex      -0.03360      0.26 0.609999999999999986677323704498121514916419983
## rx       -0.02090     33.10 0.000000064500000000000001845813541075103092481
## obstruct  0.24200      9.07 0.002599999999999999880651024852795671904459596
## perfor    0.26400      2.16 0.141999999999999987343457519273215439170598984
## adhere    0.31500     13.20 0.000284000000000000018488682806960810012242291
## nodes     0.08680    192.00 0.000000000000000000000000000000000000000000119
## differ    0.30500     20.50 0.000005859999999999999798351406343766001327822
## extent    0.57700     55.30 0.000000000000101999999999999994515315398662229
## surg      0.24600     11.70 0.000610999999999999997973842980059089313726872
## node4     0.90600    178.00 0.000000000000000000000000000000000000000136000
## etype    -0.21500     10.60 0.001129999999999999928113059155521114007569849
```

Double Checking Numerical Predictors

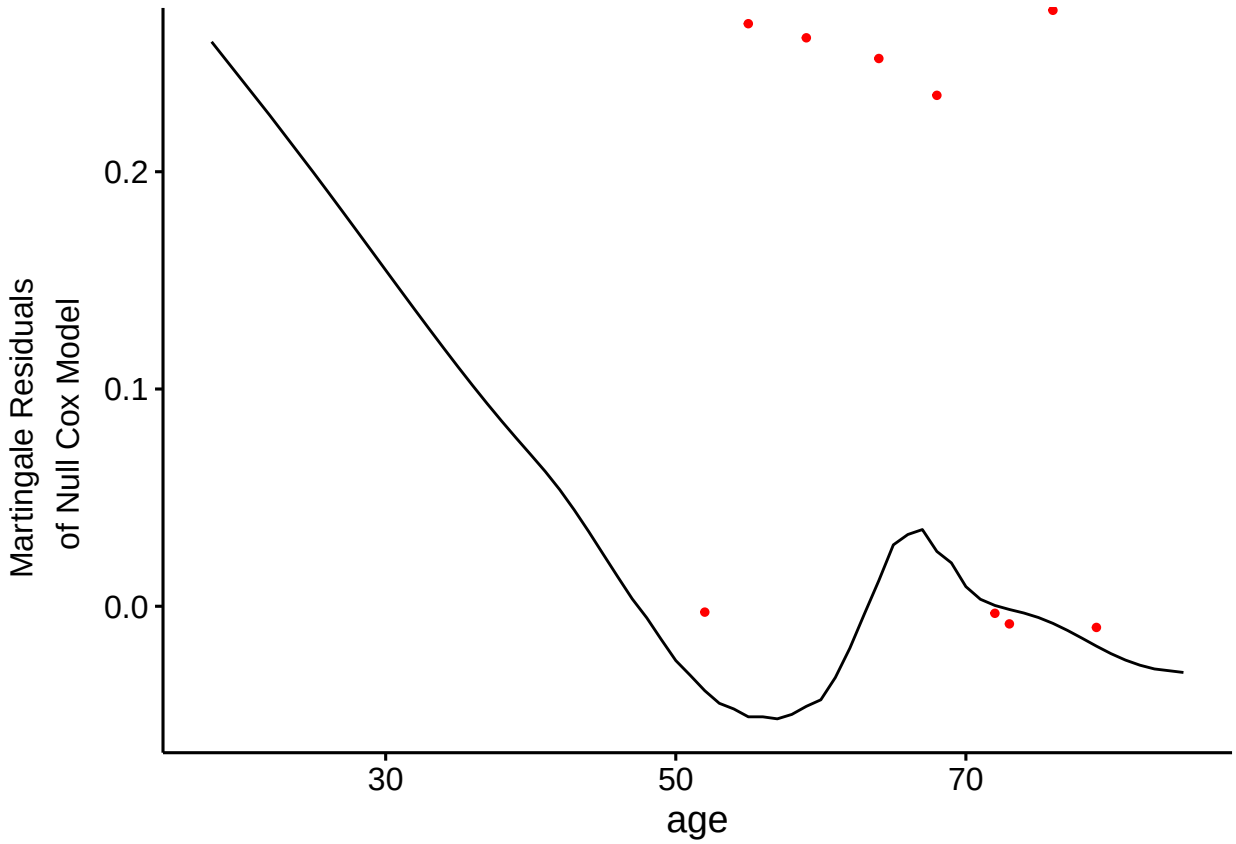
Age

```
fit <- coxph(Surv(time,status)~age,data=colon)
fit
```

```
## Call:
## coxph(formula = Surv(time, status) ~ age, data = colon)
##
##          coef exp(coef)  se(coef)      z     p
## age -0.002444  0.997559  0.002795 -0.874 0.382
##
## Likelihood ratio test=0.76  on 1 df, p=0.3831
## n= 1858, number of events= 920
```

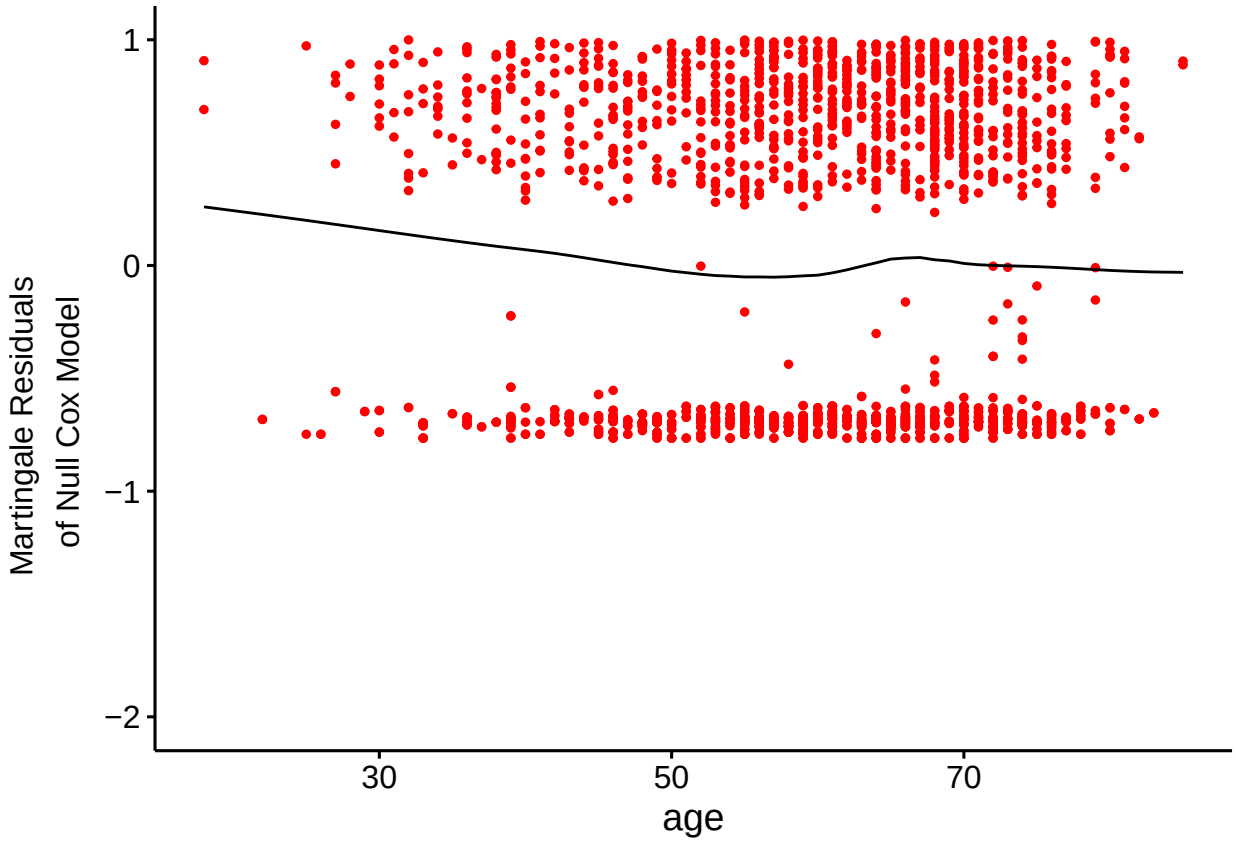
```
ggcoxfunctional(Surv(time, status) ~ age, data = colon)
```

```
## Warning: arguments formula is deprecated; will be removed in the next version;
## please use fit instead.
```




```
ggcoxfunctional(Surv(time,status)~age,data=colon,ylim=c(-2,1))
```

```
## Warning: arguments formula is deprecated; will be removed in the next version;
## please use fit instead.
```




```
# Change Z.range to your desired values
Z.range<- seq(45,55,1) # possible age values
mincutoff <- 0
minaic <- 1e10
for(cutoff in Z.range){
  # Change time, status, age, colon to your data set and variable
fit.temp <- coxph(Surv(time,status)~age*I(age>cutoff),data=colon)
aic <- AIC(fit.temp)
cat("cutoff: Z >", cutoff, "; AIC =", aic, "\n")
if(aic < minaic){
  mincutoff <- cutoff
  minaic <- aic
}
}
```

```
## cutoff: Z > 45 ; AIC = 13209.95
## cutoff: Z > 46 ; AIC = 13210.75
## cutoff: Z > 47 ; AIC = 13209.4
## cutoff: Z > 48 ; AIC = 13210.31
## cutoff: Z > 49 ; AIC = 13210.44
## cutoff: Z > 50 ; AIC = 13210.4
## cutoff: Z > 51 ; AIC = 13209.83
## cutoff: Z > 52 ; AIC = 13210.09
## cutoff: Z > 53 ; AIC = 13210.1
## cutoff: Z > 54 ; AIC = 13210.1
## cutoff: Z > 55 ; AIC = 13209.81
```

```
cat("optimal cutoff: Z >", mincutoff, "; AIC =", minaic, "\n")
```

```
## optimal cutoff: Z > 47 ; AIC = 13209.4
```

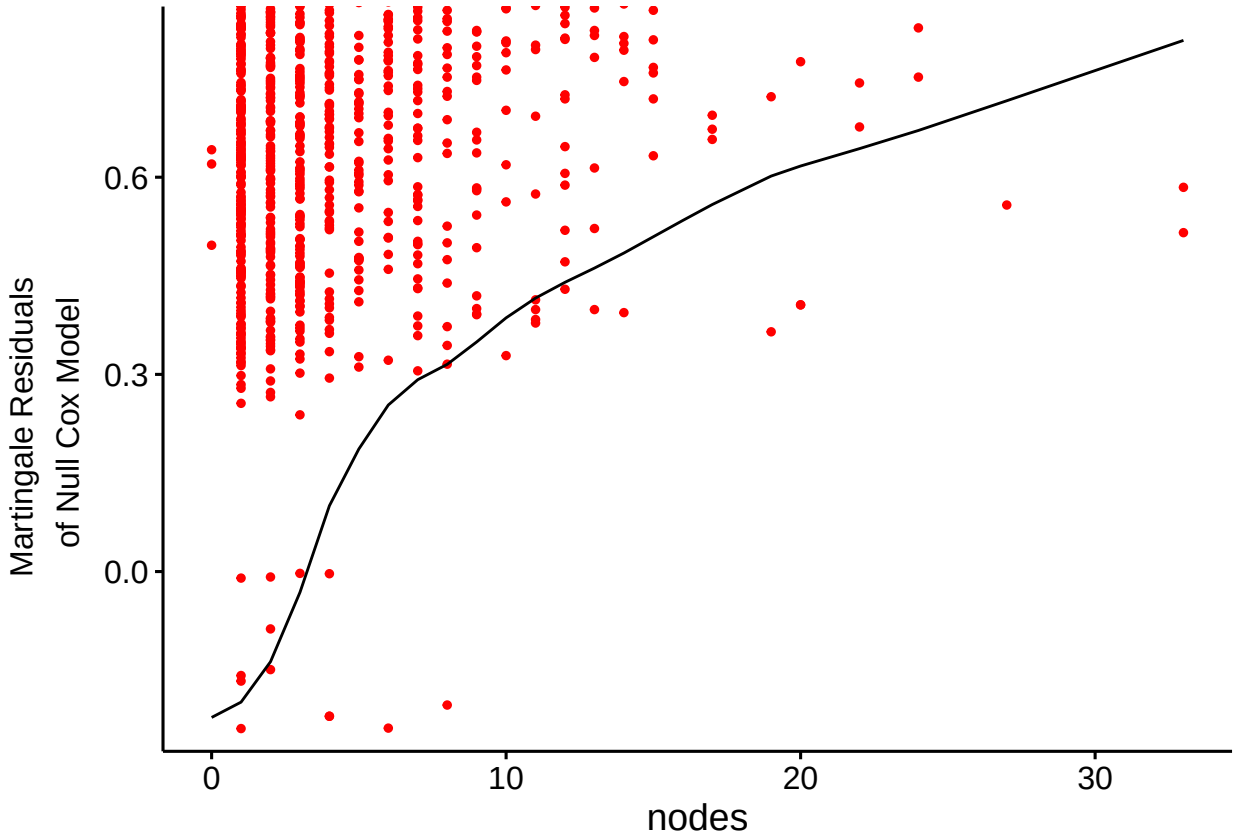
```
fit2 <- coxph(Surv(time,status)~age*I(age>47),data=colon)
AIC(fit,fit2)
```

```
##      df      AIC
## fit   1 13213.12
## fit2  3 13209.40
```

nodes

```
 I <- which(is.na(colon$nodes)) # indices where nodes is NA
colonTemp <- colon[-I,]
ggcoxfunctional(Surv(time,status)~nodes,data=colonTemp)
```

```
## Warning: arguments formula is deprecated; will be removed in the next version;
## please use fit instead.
```



```r
fit1 <- coxph(Surv(time,status)~nodes,data=colon)
fit2 <- coxph(Surv(time,status)~sqrt(nodes),data=colon)
fit3 <- coxph(Surv(time,status)~log(nodes+1),data=colon)
fit1$coefficients; fit2$coefficients; fit3$coefficients;
```

```
##      nodes
## 0.08677759
```

```
## sqrt(nodes)
##   0.5106492
```

```
## log(nodes + 1)
##      0.7198119
```

```r
anova(fit1,fit2,fit3)
```

```
## Analysis of Deviance Table
##  Cox model: response is  Surv(time, status)
##  Model 1: ~ nodes
##  Model 2: ~ sqrt(nodes)
##  Model 3: ~ log(nodes + 1)
##    loglik   Chisq Df P(>|Chi|)
## 1 -6357.4
## 2 -6340.9 33.0083  0 < 2.2e-16 ***
## 3 -6336.4  9.0568  0 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(fit1,fit2,fit3)
```

```
##      df      AIC
## fit1  1 12716.82
## fit2  1 12683.81
## fit3  1 12674.75
```

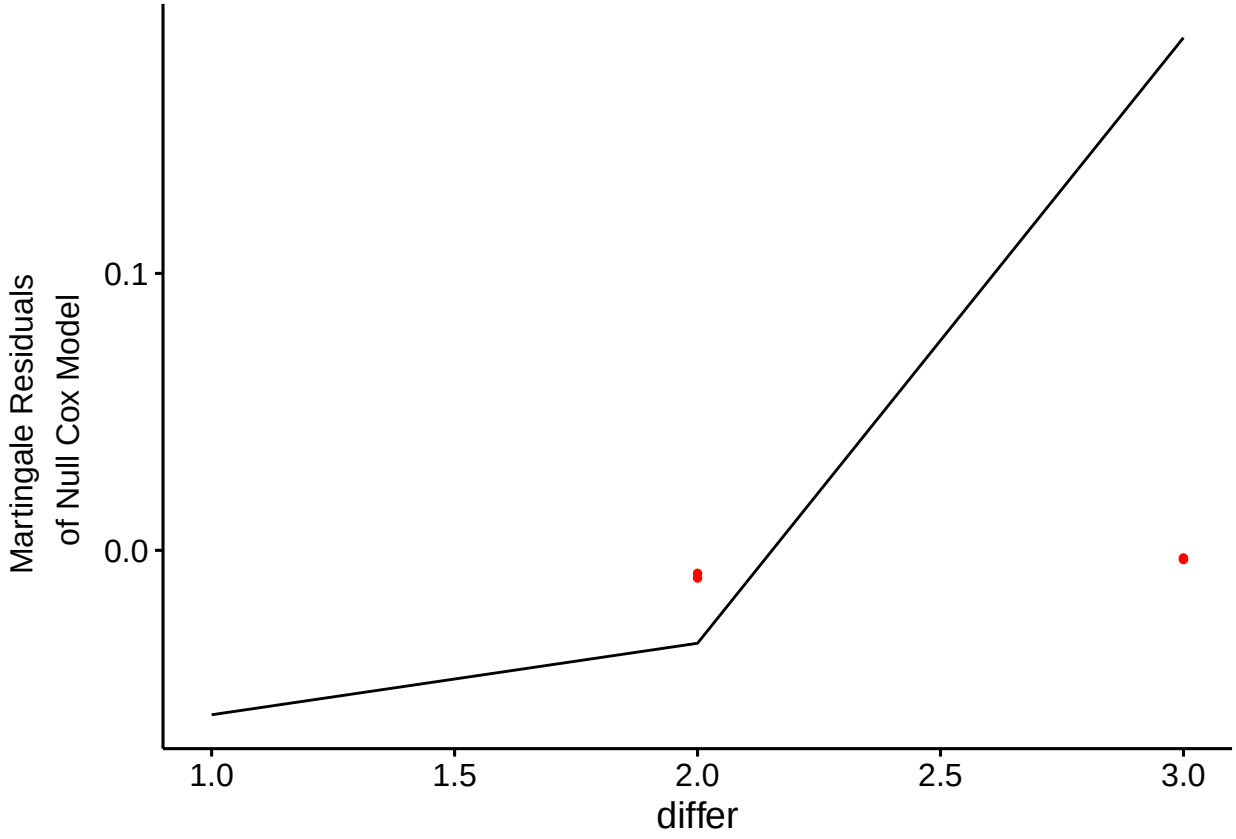
```r
fit3
```

```
## Call:
## coxph(formula = Surv(time, status) ~ log(nodes + 1), data = colon)
##
##                    coef exp(coef) se(coef)     z      p
## log(nodes + 1) 0.71981   2.05405  0.05205 13.83 <2e-16
##
## Likelihood ratio test=175.2  on 1 df, p=< 2.2e-16
## n= 1822, number of events= 897
##    (36 observations deleted due to missingness)
```

differ

```
I <- which(is.na(colon$differ))
ggcoxfunctional(Surv(time, status) ~ differ, data = colon[-I,])
```

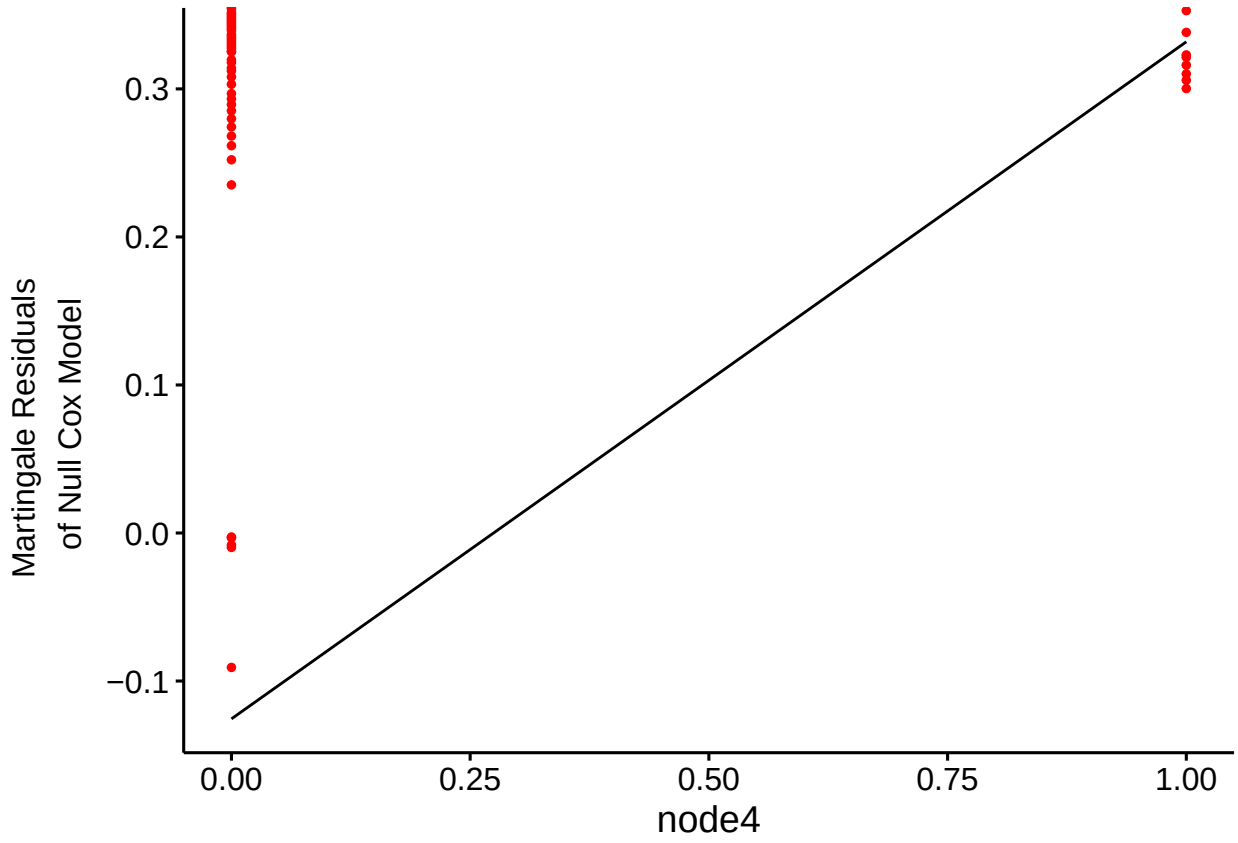
```
## Warning: arguments formula is deprecated; will be removed in the next version;
## please use fit instead.
```



node4

```
ggcoxfunctional(Surv(time, status) ~ node4, data = colon)
```

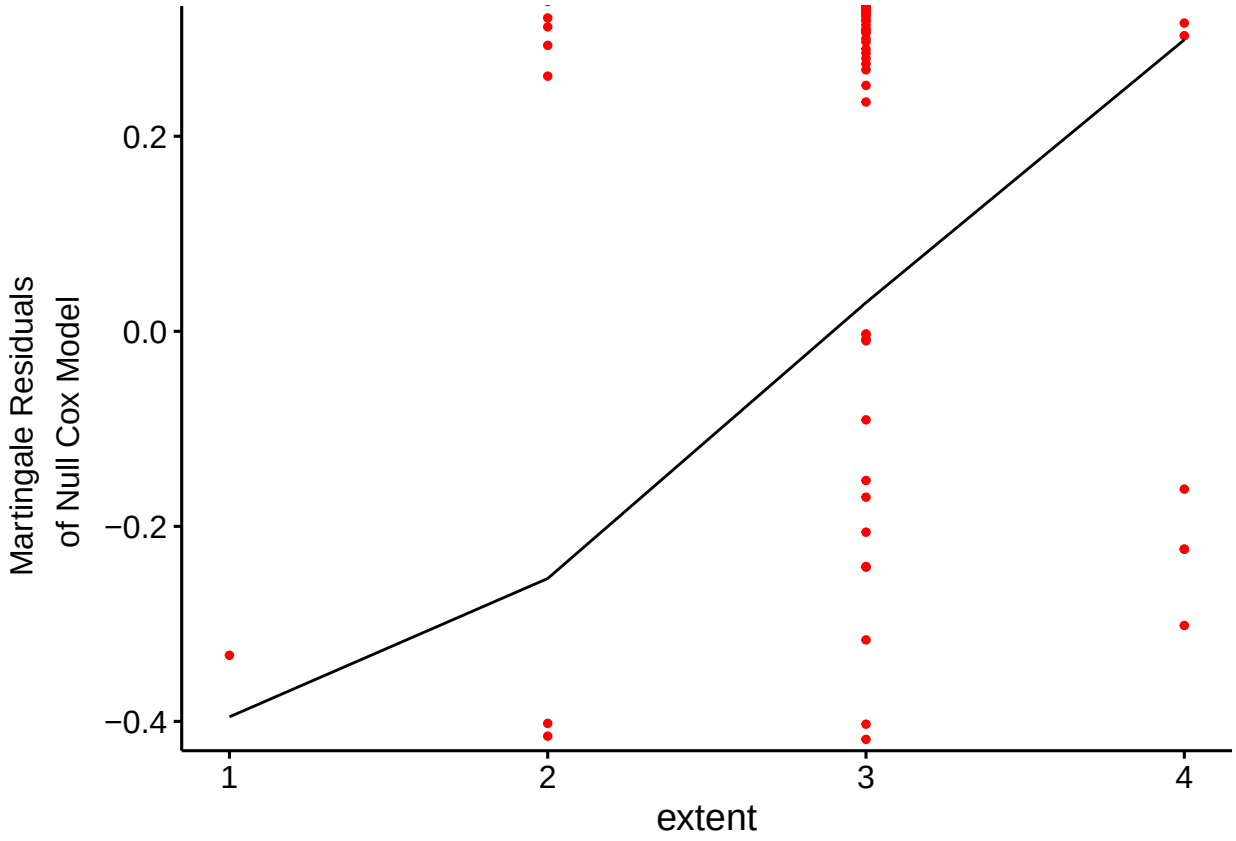
```
## Warning: arguments formula is deprecated; will be removed in the next version;
## please use fit instead.
```



extent

```
ggcoxfunctional(Surv(time, status) ~ extent, data = colon)
```

```
## Warning: arguments formula is deprecated; will be removed in the next version;
## please use fit instead.
```



The Model Thus Far

```
fit <- coxph(Surv(time,status)~age*I(age>47)+log(nodes+1)+rx+sex+obstruct+differ+surg+adhere,data=colon)
```

Removing Predictors, Testing Assumptions

```
summary(fit)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ age * I(age > 47) + log(nodes +
##     1) + rx + sex + obstruct + differ + surg + adhere, data = colon)
##
##   n= 1776, number of events= 876
##    (82 observations deleted due to missingness)
##
##                          coef exp(coef)  se(coef)      z Pr(>|z|)
## age                  -0.002749  0.997255  0.013230 -0.208  0.83542
## I(age > 47)TRUE      -0.671740  0.510819  0.608832 -1.103  0.26989
## log(nodes + 1)        0.706695  2.027280  0.053259 13.269  < 2e-16 ***
## rxLev                -0.037738  0.962965  0.079511 -0.475  0.63505
## rxLev+5FU            -0.426832  0.652573  0.086273 -4.947 7.52e-07 ***
```

```
## sex                 -0.086973  0.916702  0.068049 -1.278  0.20122
## obstruct             0.219280  1.245180  0.083860  2.615  0.00893 **
## differ               0.172381  1.188131  0.069707  2.473  0.01340 *
## surg                 0.239118  1.270129  0.074282  3.219  0.00129 **
## adhere               0.220935  1.247242  0.091318  2.419  0.01555 *
## age:I(age > 47)TRUE  0.010871  1.010931  0.014027  0.775  0.43831
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## age                    0.9973     1.0028    0.9717    1.0235
## I(age > 47)TRUE        0.5108     1.9576    0.1549    1.6846
## log(nodes + 1)         2.0273     0.4933    1.8263    2.2503
## rxLev                  0.9630     1.0385    0.8240    1.1254
## rxLev+5FU              0.6526     1.5324    0.5511    0.7728
## sex                    0.9167     1.0909    0.8022    1.0475
## obstruct               1.2452     0.8031    1.0565    1.4676
## differ                 1.1881     0.8417    1.0364    1.3621
## surg                   1.2701     0.7873    1.0980    1.4692
## adhere                 1.2472     0.8018    1.0428    1.4917
## age:I(age > 47)TRUE    1.0109     0.9892    0.9835    1.0391
##
## Concordance= 0.656  (se = 0.009 )
## Likelihood ratio test= 239.7  on 11 df,   p=<2e-16
## Wald test            = 255.4  on 11 df,   p=<2e-16
## Score (logrank) test = 262.1  on 11 df,   p=<2e-16
```

Testing the Cox PH Assumption

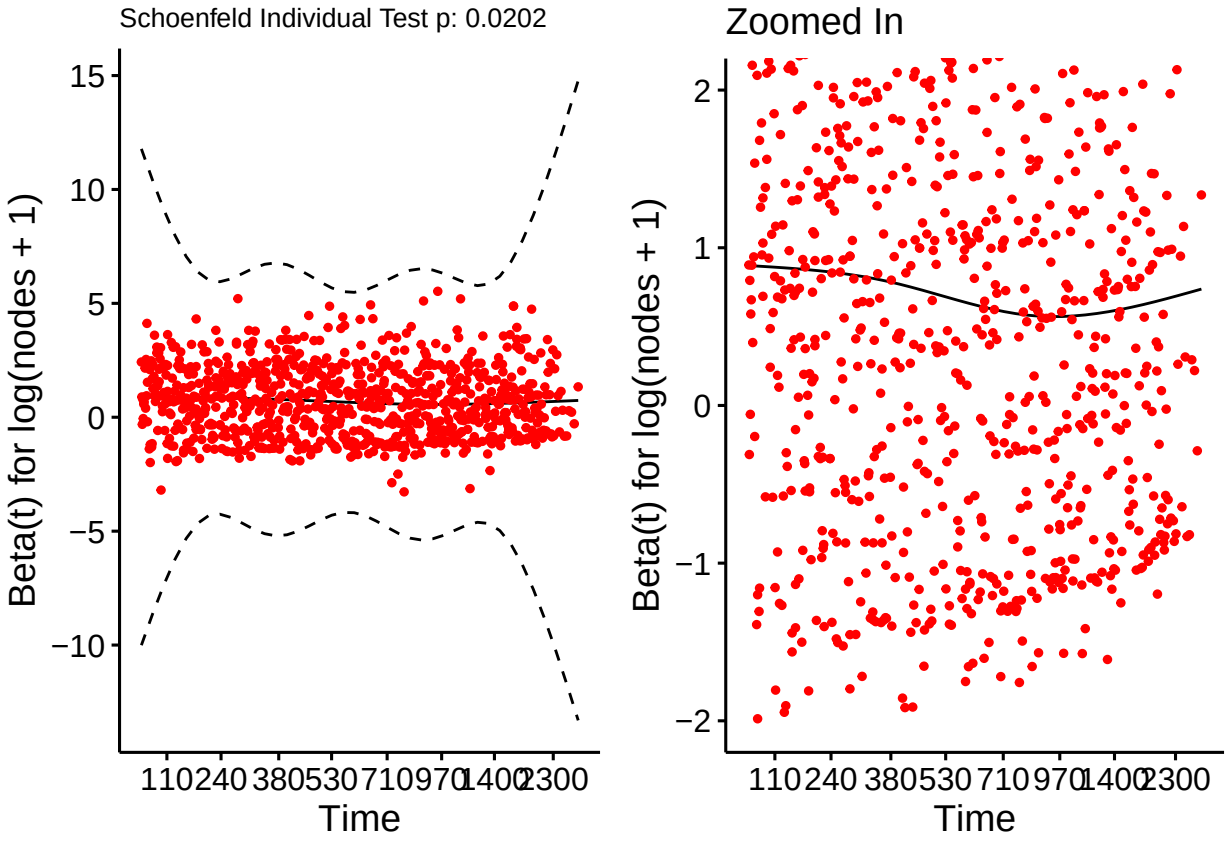
```
test.fit <- cox.zph(fit)
test.fit
```

```
##                   chisq df       p
## age              0.0659  1  0.7974
## I(age > 47)      1.8791  1  0.1704
## log(nodes + 1)   5.3935  1  0.0202
## rx               1.7045  2  0.4265
## sex              1.9286  1  0.1649
## obstruct         9.0392  1  0.0026
## differ          20.5634  1 5.8e-06
## surg             0.7176  1  0.3969
## adhere           0.5308  1  0.4663
## age:I(age > 47)  1.0627  1  0.3026
## GLOBAL          45.3342 11 4.2e-06
```

nodes

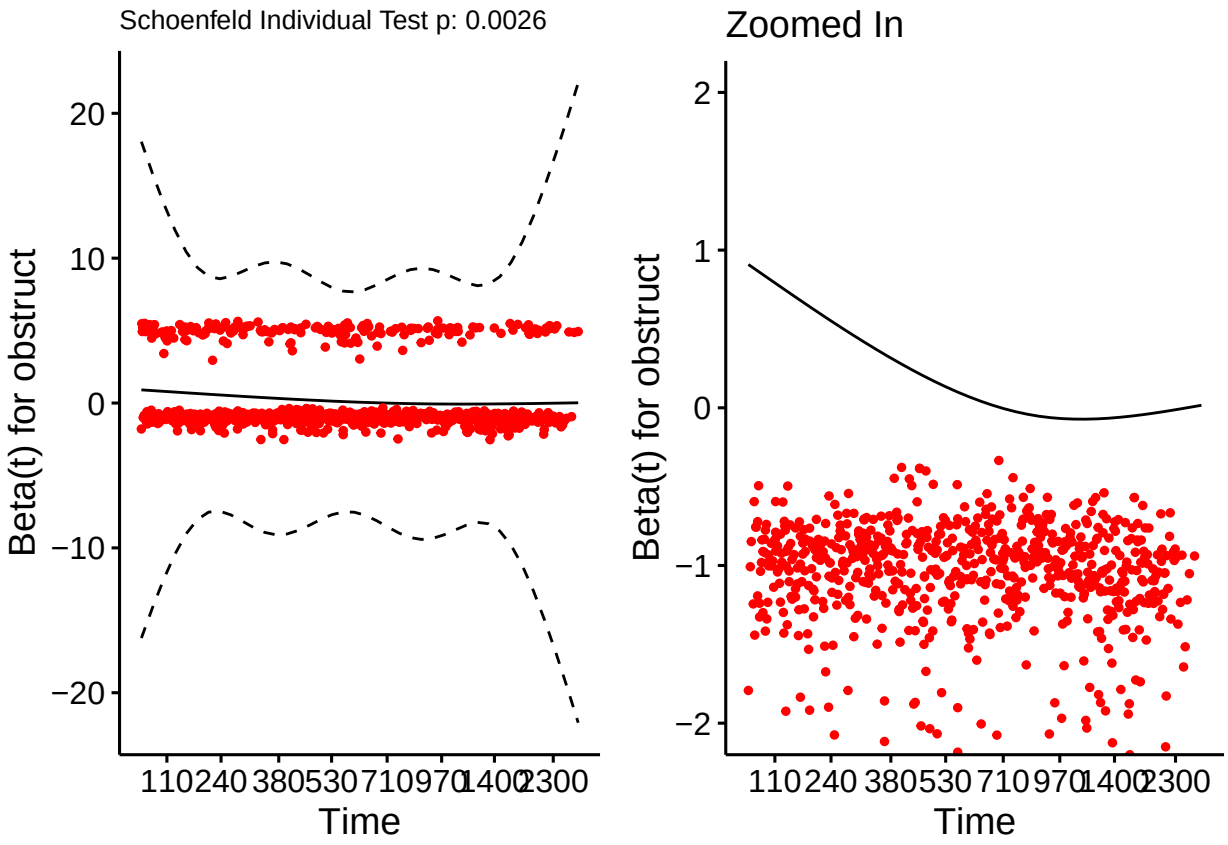
```
par(mfrow=c(1,2))
library(gridExtra)
plot1 <- ggcoxzph(test.fit,var="log(nodes + 1)",font.main=c(10,"plain","black"))
```

```
plot2 <- ggcoxzph(test.fit,var="log(nodes + 1)",ylim=c(-2,2),font.main=c(14,"plain","black"),main="Zoom
 grid.arrange(plot1[[1]],plot2[[1]],ncol=2)
```



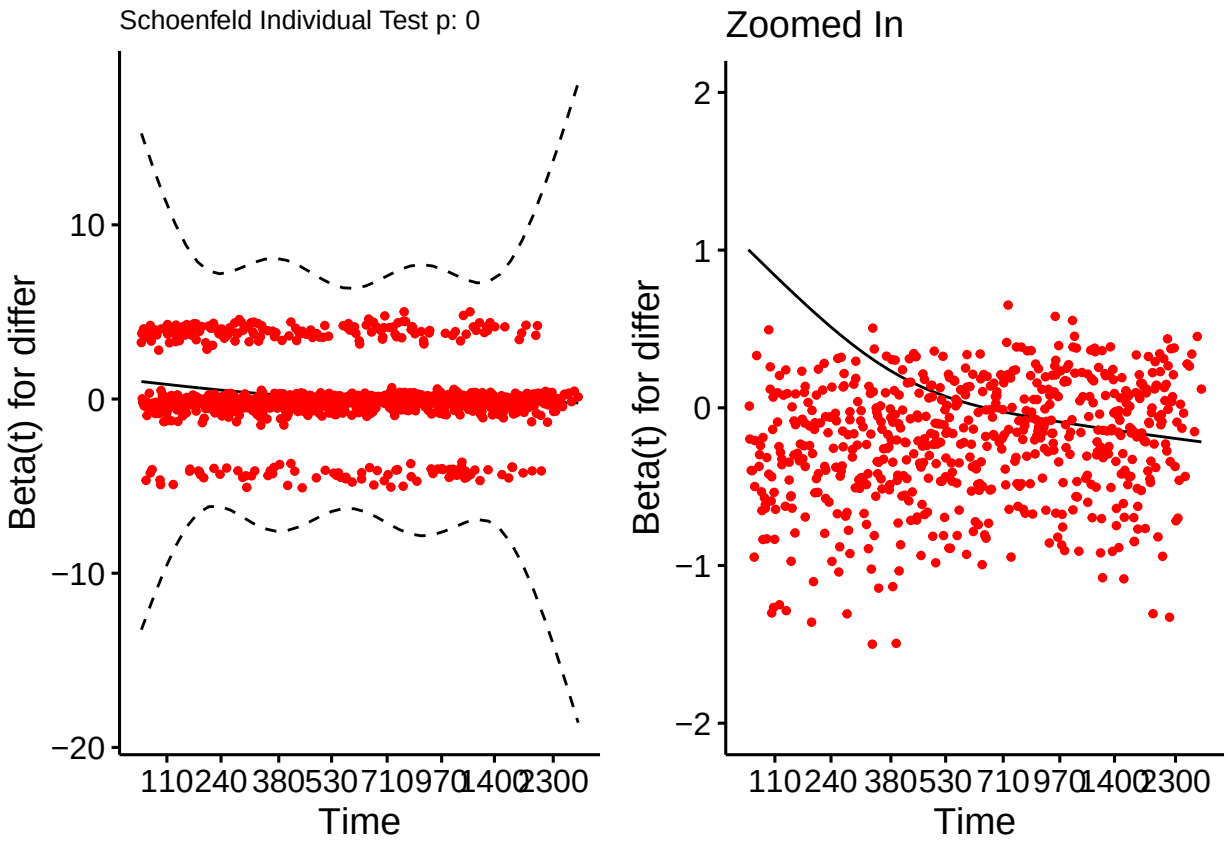
Obstruct

```
plot1 <- ggcoxzph(test.fit,var="obstruct",font.main=c(10,"plain","black"))
plot2 <- ggcoxzph(test.fit,var="obstruct",ylim=c(-2,2),font.main=c(14,"plain","black"),main="Zoomed In"
grid.arrange(plot1[[1]],plot2[[1]],ncol=2)
```



differ

```
plot1 <- ggcoxzph(test.fit,var="differ",font.main=c(10,"plain","black"))
plot2 <- ggcoxzph(test.fit,var="differ",ylim=c(-2,2), font.main=c(14,"plain","black"),main="Zoomed In")
grid.arrange(plot1[[1]],plot2[[1]],ncol=2)
```



Removing Covariates

differ

```
fit1 <- coxph(Surv(time,status)~age*I(age>47)+log(nodes+1)+rx+sex+surg+obstruct+I(differ>2)+adhere,data=
fit2 <- coxph(Surv(time,status)~age*I(age>47)+log(nodes+1)+rx+sex+surg+obstruct+as.factor(differ)+adhere
```

```
temp <- cox.zph(fit1)
print(temp)
```

```
##                    chisq df       p
## age               0.0918  1  0.7619
## I(age > 47)       2.0013  1  0.1572
## log(nodes + 1)    5.2887  1  0.0215
## rx                1.8324  2  0.4000
## sex               2.0068  1  0.1566
## surg              0.6124  1  0.4339
## obstruct          9.0134  1  0.0027
## I(differ > 2)    29.2062  1 6.5e-08
## adhere            0.4861  1  0.4857
## age:I(age > 47)   1.1525  1  0.2830
## GLOBAL           53.0417 11 1.8e-07
```

```
temp <- cox.zph(fit2)
print(temp)
```

```
##                      chisq df       p
## age                 0.0876  1  0.7673
## I(age > 47)         1.9937  1  0.1580
## log(nodes + 1)      5.3034  1  0.0213
## rx                  1.8587  2  0.3948
## sex                 2.0167  1  0.1556
## surg                0.5908  1  0.4421
## obstruct            9.0175  1  0.0027
## as.factor(differ)  29.5834  2 3.8e-07
## adhere              0.4916  1  0.4832
## age:I(age > 47)     1.1413  1  0.2854
## GLOBAL             53.2500 12 3.7e-07
```

dealing with obstruct

```
fit1 <- coxph(Surv(time,status)~age*I(age>47)+log(nodes+1)+rx+sex+obstruct:nodes+surg+adhere,data=colon)
temp <- cox.zph(fit1)
print(temp)
```

```
##                     chisq df     p
## age               0.00154  1 0.969
## I(age > 47)       1.52307  1 0.217
## log(nodes + 1)    4.68024  1 0.031
## rx                1.07976  2 0.583
## sex               2.58251  1 0.108
## surg              0.65195  1 0.419
## adhere            0.46210  1 0.497
## age:I(age > 47)   0.73258  1 0.392
## obstruct:nodes    3.13963  1 0.076
## GLOBAL           15.18098 10 0.126
```

dealing with nodes

```
fit2 <- coxph(Surv(time,status)~age*I(age>47)+sqrt(nodes)+rx+sex+obstruct:nodes+surg+adhere,data=colon)
temp <- cox.zph(fit2)
print(temp)
```

```
##                     chisq df    p
## age               0.00119  1 0.97
## I(age > 47)       1.63797  1 0.20
## sqrt(nodes)       1.52597  1 0.22
## rx                1.12663  2 0.57
## sex               2.87802  1 0.09
## surg              0.59927  1 0.44
## adhere            0.38386  1 0.54
```

```
## age:I(age > 47)  0.77590  1 0.38
## obstruct:nodes   2.33470  1 0.13
## GLOBAL          12.60068 10 0.25
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.