

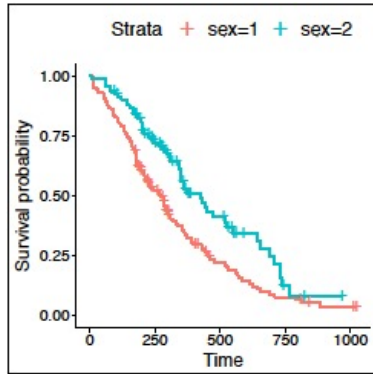
# Creating Survival Plots

## Informative and Elegant with *survminer*

### Curvas de supervivencia

La función **ggsurvplot()** crea gráficas de ggplot2 a partir de objetos **survfit**.

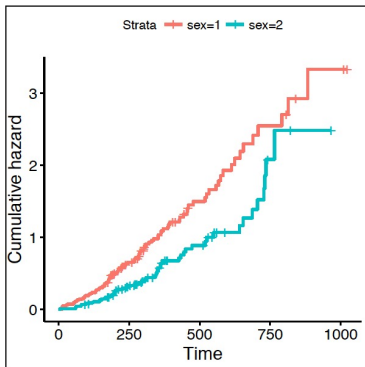
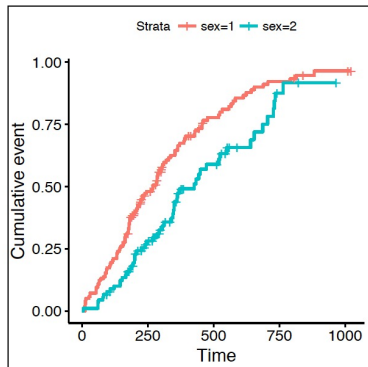
```
library("survival")
fit <- survfit(Surv(time,status)
  ~ sex, data = lung)
class(fit)
## [1] "survfit"
```



```
library("survminer")
ggsurvplot(fit, data = lung)
```

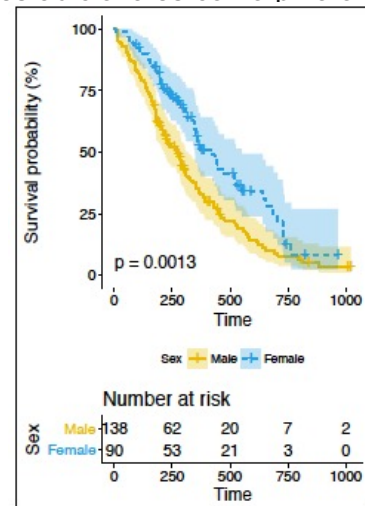
Usa el argumento **fun** para establecer la transformación de la curva de supervivencia. Por ejemplo, **"event"** para eventos acumulativos, **"cumhaz"** para la función de riesgo acumulado o **"pct"** para la probabilidad de supervivencia en porcentaje.

```
ggsurvplot(fit, data = lung, fun = "event")
ggsurvplot(fit, data = lung, fun = "cumhaz")
```



Con muchos de los parámetros de los gráficos de supervivencia tienes control total sobre cómo se ven; la posición y el contenido de la leyenda; anotaciones adicionales como p-valor, título y subtítulo.

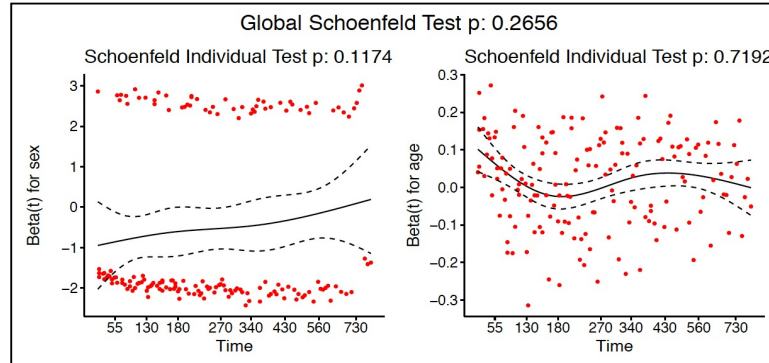
```
ggsurvplot(fit, data = lung,
  conf.int = TRUE,
  pval = TRUE,
  fun = "pct",
  risk.table = TRUE,
  size = 1,
  linetype = "strata",
  palette = c("#E7B800", "#2E9FDF"),
  legend = "bottom",
  legend.title = "Sex",
  legend.labs = c("Male", "Female"))
```



### Diagnósticos del modelo de Cox

La función **cox.zph()** del paquete **survival** puede ser utilizada para probar el supuesto de riesgos proporcionales para un ajuste del modelo de regresión de Cox. La verificación gráfica de esta suposición se puede realizar con la función **ggcoxzph()** del paquete **survminer**. Para cada covariable, produce gráficos con residuos de Schoenfeld escalados contra el tiempo.

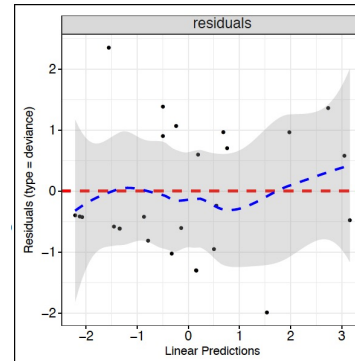
```
library("survival")
fit <- coxph(Surv(time, status) ~ sex + age, data = lung)
ftest <- cox.zph(fit)
ggcoxzph(ftest)
```



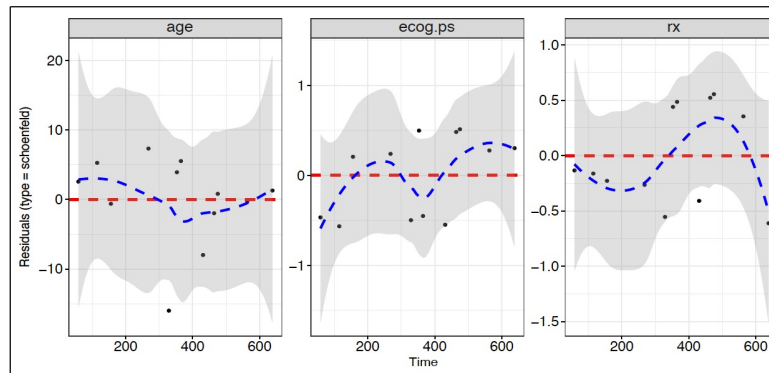
La función **ggcoxdiagnostics()** grafica diferentes tipos de residuales en función del tiempo, predictor lineal u observación por id. El tipo de residual se selecciona con el argumento **tipo**. Posibles valores "martingale", "deviance", "score", "schoenfeld", "dfbeta", "dfbetas", and "scaledsch". El argumento **ox.scale** define que debe ser representado en el eje OX. Los posibles valores son "linear.predictions", "observation.id", "time". Los argumentos lógicos **hline** and **sline** pueden ser usados para añadir una línea horizontal o una línea suavizada a la gráfica.

```
library("survival")
library("survminer")
fit <- coxph(Surv(futime, fustat) ~
  age + ecog.ps + rx, data=ovarian)
```

```
ggcoxdiagnostics(fit,
  type = "deviance",
  ox.scale = "linear.predictions")
```



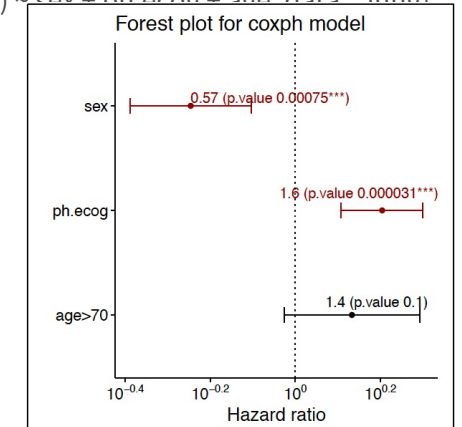
```
ggcoxdiagnostics(fit,
  type = "schoenfeld",
  ox.scale = "time")
```



### Sumario del modelo de Cox

La función **ggforest()** del paquete **survminer** crea un diagrama de bosque para un ajuste del modelo de regresión de Cox. Las estimaciones de la razón de riesgo junto con los intervalos de confianza y los valores p se trazan para cada variable.

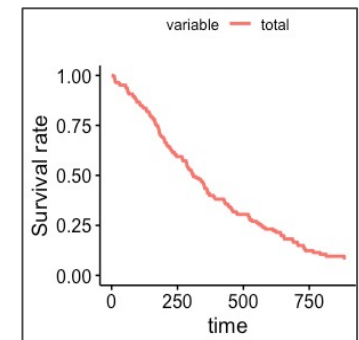
```
library("survival")
library("survminer")
lung$age <- ifelse(lung$age > 70, ">70", "<= 70")
fit <- coxph(Surv(time, status) ~ sex + ph.ecog + age, data = lung)
ggforest(fit)
```



La función **ggadjustedcurves()** del paquete **survminer** grafica curvas ajustadas de supervivencia para un modelo de riesgos proporcionales de Cox.

Las curvas de supervivencia ajustadas enseñan cuán un seleccionado factor influencia la supervivencia estimada por un modelo de Cox. Nótese que estas curvas difieren de las curvas de Kaplan Meier estimadas ya que presentan supervivencia esperada basada en un determinado modelo de Cox.

```
library("survival")
library("survminer")
fit <- coxph(Surv(time, status) ~
  ph.ecog + age + sex,
  data = lung)
ggadjustedcurves(fit, data = lung)
```



Nótese que la opción para representar curvas individuales sin agrupamiento no está disponible en la nueva función **ggadjustedcurves**

```
fit <- coxph(Surv(time, status) ~
  ph.ecog + age + strata(sex),
  data = lung)
ggadjustedcurves(fit, data = lung)
```

