

```
# APPENDIX
```

```
# READ IN DATASET
```

```
data <- read.table("assignOct3-1.txt", header=TRUE, row.names=1)
# check read
head(data)
# check dimensions
dim(data)
# attach
attach(data)
# structure check
str(data)
```

```
# QUESTION 1
```

```
# events
table(data$Event)

# age
summary(data$Age)

# menopause
table(data$MenopauseStatus)
table(data$MenopauseStatus, data$Event)

# hormone therapy
table(data$HormoneTherapy)
table(data$HormoneTherapy, data$Event)

# tumour size
summary(data$TumourSize)

# tumour grade
table(data$TumourGrade)
table(data$TumourGrade, data$Event)

# Median survival
library(survival)
survfit(Surv(data$Survival, data$Event) ~ 1)
summary(data$Survival)
```

```
# QUESTION 2
```

```
# Cox proportional hazards regression analysis
model_full <- coxph(Surv(Survival, Event) ~ Age + MenopauseStatus +
                    HormoneTherapy + TumourSize + TumourGrade,
                    data = data)

model_full
# stepwise selection
model_final <- step(model_full)

# model summaries
summary(model_final)

# AIC values check (lower is better fit)
AIC(model_full)
AIC(model_final)

# visuals
# km tumour grade
library(survival)
KM.tumourgrade <- survfit(Surv(Survival, Event) ~ TumourGrade,
                          data = data)
diff.test <- survdiff(Surv(Survival, Event) ~ TumourGrade,
                      data = data)
p.value.TG <- format(diff.test$p, scientific = TRUE, digits = 2)
```

```

plot(KM.tumourgrade, main = "Tumour Grade", xlab = "Time (Days)",
     ylab = "Survival Probability", col = c("pink", "red", "darkred"),
     lty = 1)
legend("bottomleft", title = "Tumour Grade", legend = c("1", "2", "3"),
     col = c("pink", "red", "purple"), lty = 1, cex = 0.8)
text(1500, 0.5, paste("p =", p.value.TG), cex = 0.8)

# km menopause
KM.menopause <- survfit(Surv(Survival, Event) ~ MenopauseStatus,
                        data = data)
diff.test <- survdiff(Surv(Survival, Event) ~ MenopauseStatus,
                      data = data)
p.value.MP <- format(diff.test$p, scientific = TRUE, digits = 2)

plot(KM.menopause, main = "Menopause Status", xlab = "Time (Days)",
     ylab = "Overall Survival Probability", col = c("red", "green"),
     lty = 1)
legend("bottomleft", title = "Menopause Status", c("Post", "Pre"),
     lty = 1,
     col = c("red", "green"), cex = 0.8)
text(1500, 0.5, paste("p =", p.value.MP), cex = 0.8)

# km tumour size
# break into groups
data$TumourSizeGroup <- cut(data$TumourSize,
                            c(0, 20, 40, 60, 80, Inf),
                            labels = c("<20", "21-40", "41-60",
                                       "61-80", ">81"))

# create curves
my.KMest <- survfit(Surv(Survival, Event) ~ TumourSizeGroup,
                    data = data)

# plot
plot(my.KMest, main = "Tumour Size", xlab = "Time (Days)",
     ylab = "Overall Survival Probability",
     col = c("darkgreen", "lightgreen", "yellow", "orange", "red"),
     lty = 1)

# add legend
legend("bottomleft", title = "Tumour Size",
     c("<20", "21-40", "41-60", "61-80", ">81"),
     lty = 1,
     col = c("darkgreen", "lightgreen", "yellow", "orange", "red"),
     cex = 0.8)

# add p-value
fit <- survdiff(Surv(Survival, Event) ~ TumourSizeGroup, data = data)
p.value <- format(round(summary(fit)$chisq["pvalue"], 4), nsmall = 4)
legend("bottomright", paste0("p-value: ", p.value), cex = 0.8,)

# QUESTION 3

# split data into pre-menopausal and post-menopausal groups
pre_meno <- subset(data, data$MenopauseStatus == 1)
post_meno <- subset(data, data$MenopauseStatus == 2)

# compare behavior of age in pre- and post-menopausal patients
# measures of central tendency
mean(pre_meno$Age)
mean(post_meno$Age)
# boxplot
boxplot(pre_meno$Age, post_meno$Age,
        main = "Age Distribution by MenopauseStatus",
        names = c("Pre-Menopausal", "Post-Menopausal"))

# do older pre menopausal patients have a lower or higher
# survival than younger post menopausal patients?
# filter by age and menopausal status
oldpremenopausal <- subset(data,
                            MenopauseStatus == 1 & Age > 45)

```

```

        youngpostmenopausal <- subset(data,
                                      MenopauseStatus == 2 & Age < 59)
    # check normality
    shapiro_test_oldpremenopausal <-
shapiro.test(oldpremenopausal$Survival)
    shapiro_test_youngpostmenopausal <-
shapiro.test(youngpostmenopausal$Survival)
    # print Shapiro-Wilk
    cat("Shapiro-Wilk test for normality of old premenopausal patients'
survival data: p =", shapiro_test_oldpremenopausal$p.value, "\n")
    cat("Shapiro-Wilk test for normality of young postmenopausal
patients' survival data: p =", shapiro_test_youngpostmenopausal$p.value, "\n")
    # compare survival
    premenopausal_survival <- oldpremenopausal$Survival
    postmenopausal_survival <- youngpostmenopausal$Survival
    # perform t-test
    t_test_result <- t.test(premenopausal_survival,
                           postmenopausal_survival,
                           alternative = "less")

    # print results of t-test
    cat("t-test for difference in survival between older premenopausal
and younger postmenopausal patients: p =", t_test_result$p.value, "\n")

# UNIVARIATE

# pre-menopausal group modeling of age and survival
premenopausal_model <- coxph(Surv(Survival, Event) ~ Age,
                             data = pre_meno)
summary(premenopausal_model)

# post-menopausal group modeling of age and survival
postmenopausal_model <- coxph(Surv(Survival, Event) ~ Age,
                              data = post_meno)
summary(postmenopausal_model)

# Wilcox
wilcox.test(pre_meno$Age ~ pre_meno$Event) # Wilcoxon rank-sum test for
age and survival
wilcox.test(post_meno$Age ~ post_meno$Event) # Wilcoxon rank-sum test for
age and survival

# visuals
# KM curves for pre-menopausal and post-menopausal age groups
km_pre <- survfit(Surv(Survival, Event) ~
                  cut(Age, breaks=c(0, 40, 50, 60,
                                   max(pre_meno$Age))),
                  data=pre_meno)
km_post <- survfit(Surv(Survival, Event) ~
                  cut(Age, breaks=c(0, 40, 50, 60,
                                   max(post_meno$Age))),
                  data=post_meno)

# plot
plot(km_pre, col=c("red", "blue", "green", "orange"),
     xlab="Time (days)", ylab="Survival Probability",
     main="Kaplan-Meier Curves by Age Group - Pre-Menopausal")
# legend
legend("bottomleft",
      legend = c("<40", "41-50", "51-60", ">60"),
      col=c("red", "blue", "green", "orange"), lty=1, cex=0.8)

# plot
plot(km_post, col=c("red", "blue", "green", "orange"),
     xlab="Time (days)", ylab="Survival Probability",
     main="Kaplan-Meier Curves by Age Group - Post-Menopausal")
# legend
legend("bottomleft", legend = c("<40", "41-50", "51-60", ">60"),
      col=c("red", "blue", "green", "orange"), lty=1, cex=0.8)

# plot both
plot(km_pre, col=c("red", "blue", "green", "orange"),

```

```

        xlab="Time (days)", ylab="Survival Probability",
        main="Kaplan-Meier Curves by Age Group - Pre-Menopausal")
lines(km_post, col=c("red","blue","green","orange"),
      lty=2) # dashed post meno lines
legend("bottomleft", legend=c("<40", "41-50", "51-60", ">60"),
      col=c("red","blue","green","orange"), lty=1)

# log-rank test for pre-menopausal group
pre_meno_test <- survdiff(Surv(Survival, Event) ~ cut
                          (Age, breaks=c(0, 40, 50, 60,
                                          max(pre_meno$Age))),
                          data=pre_meno)

pre_meno_test

# log-rank test for post-menopausal group
post_meno_test <- survdiff(Surv(Survival, Event) ~
                          cut(Age, breaks=c(0, 40, 50, 60,
                                          max(post_meno$Age))),
                          data=post_meno)

post_meno_test

```

```

# MULTIVARIATE

```

```

# Fit a multivariate Cox proportional hazards model
cox_mod <- coxph(Surv(Survival, Event) ~ Age + MenopauseStatus,
                 data = data)
# Output the results
summary(cox_mod)

# Interaction test of age and menopausal status on survival
interaction_model <- coxph(Surv(Survival, Event) ~ Age*MenopauseStatus,
                          data = data)
summary(interaction_model)

# linear regression with interaction term
summary(lm(Survival ~ Age + MenopauseStatus + age.menopause +
Event,
           data = data))
# Fit a multivariate regression model with interaction terms
lin_reg_model <- lm(cbind(Survival, Event) ~ Age * MenopauseStatus,
                   data = data)
# Extract the model summary
summary(lin_reg_model)

```

```

# QUESTION 4

```

```

# part a

```

```

# load both datasets
data1 <- read.table("assignOct3-1.txt", header = TRUE)
data2 <- read.table("assignOct4.txt", header = TRUE)

# join datasets by patient IDs
merged_data <- merge(data1, data2, by = "Ptid")

# define good and poor survival groups
good_survival <- merged_data[merged_data$Survival > 5 * 365, ]
poor_survival <- merged_data[merged_data$Survival < 365, ]

# define gene expression cols in data
gene_cols <- c("Gene1", "Gene2", "Gene3", "Gene4", "Gene5")

# wilcoxon rank sum for each gene
for (i in gene_cols) {

```

```

col_index <- which(names(merged_data) == i) # find column index of
gene in merged dataset
wilcox_result <- wilcox.test(good_survival[, col_index],
                             poor_survival[, col_index]) # perform
Wilcoxon rank sum test
p_value_wilcox <- wilcox_result$p.value

if (p_value_wilcox < 0.05) {
  print(paste(i, "has a significant difference in expression levels
(p =", p_value_wilcox, ")"))
} else {
  print(paste(i, "does not have a significant difference in
expression levels (p =", p_value_wilcox, ")"))
}
}

# apply Bonferroni correction
bonferroni_threshold <- 0.05/length(gene_cols)

# wilcoxon rank sum for each gene
for (i in gene_cols) {
  col_index <- which(names(merged_data) == i) # find column
index of gene in merged dataset
wilcox_result <- wilcox.test(good_survival[, col_index],
                             poor_survival[, col_index]) #
perform Wilcoxon rank sum test
p_value_wilcox <- wilcox_result$p.value

if (p_value_wilcox < bonferroni_threshold) {
  print(paste(i, "has a significant difference in expression
levels (p =", p_value_wilcox, ")"))
} else {
  print(paste(i, "does not have a significant difference in
expression levels (p =", p_value_wilcox, ")"))
}
}

# new column for survival group
merged_data$SurvivalGroup <- ifelse(merged_data$Survival > 5
* 365, "good", ifelse(merged_data$Survival < 365, "poor", NA))

# ANOVA for each gene with post hoc Tukey HSD
for (i in gene_cols) {
  col_index <- which(names(merged_data) == i) # find column
index of gene in merged dataset
anova_result <- aov(as.formula(paste(i, " ~
SurvivalGroup")), data = merged_data) # perform ANOVA
p_value_anova <- summary(anova_result)[[1]][["Pr(>F)"]][1]
# extract p-value from ANOVA summary

if (p_value_anova < 0.05) {
  print(paste(i, "has a significant difference in
expression levels (p =", p_value_anova, ")"))
  tukey_result <- TukeyHSD(anova_result) # perform Tukey
test
  print(tukey_result)
} else {
  print(paste(i, "does not have a significant difference in
expression levels (p =", p_value_anova, ")"))
}
}

# box plots
library(ggplot2)
for (i in gene_cols) { #loop genes
  col_index <- which(names(merged_data) == i) # find gene cols
plot_data <- data.frame( # create data frame
  expression = c(good_survival[, col_index],
                 poor_survival[, col_index]),

```

```

        survival = rep(c("Good", "Poor"), c(nrow(good_survival),
                                              nrow(poor_survival)))
    )
    boxplots <- ggplot(plot_data, aes(x = survival,
                                      y = expression,
                                      fill = survival)) + # create box
plots
    geom_boxplot() +
    labs(x = "Survival", y = i) + # axis labels
    scale_fill_manual(values = c("Good" = "green", "Poor" = "red")) +
# colours
    theme_bw()
    print(boxplots)
}

# part b

# cox models
# Gene1
cox_model_gene1 <- coxph(Surv(Survival, Event) ~ Gene1,
                        data=merged_data)
summary(cox_model_gene1)
# Gene2
cox_model_gene2 <- coxph(Surv(Survival, Event) ~ Gene2,
                        data=merged_data)
summary(cox_model_gene2)
# Gene3
cox_model_gene3 <- coxph(Surv(Survival, Event) ~ Gene3,
                        data=merged_data)
summary(cox_model_gene3)
# Gene4
cox_model_gene4 <- coxph(Surv(Survival, Event) ~ Gene4,
                        data=merged_data)
summary(cox_model_gene4)
# Gene5
cox_model_gene5 <- coxph(Surv(Survival, Event) ~ Gene5,
                        data=merged_data)
summary(cox_model_gene5)

# visuals
# boxplots
# load library
library(ggplot2)
# vector with gene names to plot
gene_names <- paste0("Gene", 1:5)
# loop
for (gene in gene_names) {
  # df of the expression values
  # sep by event
  plot_data <- data.frame(
    expression = c(merged_data[merged_data$Event == 0, gene],
                  merged_data[merged_data$Event == 1, gene]),
    survival = rep(c("Alive", "Dead"), c(sum(merged_data$Event ==
0),
1)))
    )
    # boxplot for current gene
    boxplots <- ggplot(plot_data, aes(x = survival, y = expression,
fill = survival)) +
    geom_boxplot() +
    labs(x = "Event (0 = Alive, 1 = Dead)", y = gene) + # label
the axes
    scale_fill_manual(values = c("Alive" = "green", "Dead" =
"red")) + # set colours
    theme_bw() # set plot theme
    print(boxplots) # print plot
}

```

