Multivar_HW4

Matthew Stoebe

2025-04-27

Question 1

With the information provided,

$$\begin{aligned}\mathrm{Cov}(X) &= \mathrm{Cov}(\lambda F + e) \\ &= \lambda\lambda^\top \mathrm{Var}(F) + \Psi \\ &= \lambda\lambda^\top + \Psi.\end{aligned}$$

For i != j

$$= (\lambda\lambda^\top)_{ij} = \lambda_i\lambda_j.$$

Hence the i-th row ignoring its diagonal element is

$$\left(\lambda_i\lambda_1,\ \lambda_i\lambda_2,\ \ldots,\ \lambda_i\lambda_p\right) = \lambda_i\,\lambda^\top,$$

which is the vector $\lambda^\top$ scaled by the constant λ_i.
Therefore every off-diagonal row of Σ_X is proportional to every other, as required.

Question 2

a

```
library(psych)
library(stats)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x ggplot2::%+%()   masks psych::%+%()
## x ggplot2::alpha() masks psych::alpha()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
dat_raw <- read.table("Data/testData.txt", header = TRUE,
                      stringsAsFactors = FALSE)

tests <- dat_raw[, 1:6]
colnames(tests) <- paste0("T", 1:6)

R      <- cor(tests)
eigval <- eigen(R)$values
prop   <- eigval / sum(eigval)

print(round(cbind(eigval, prop, cumsum(prop)), 3),
      quote = FALSE,
      col.names = c("eigen", "prop", "cumprop"))
```

```
##      eigval  prop
## [1,]  1.775 0.296 0.296
## [2,]  1.354 0.226 0.522
## [3,]  1.073 0.179 0.700
## [4,]  0.815 0.136 0.836
## [5,]  0.531 0.088 0.925
## [6,]  0.452 0.075 1.000
```

```r
m <- 3


fa_raw <- principal(tests,
                    nfactors = m,
                    rotate   = "none",
                    scores   = TRUE)

cat("\n=== (a)  Unrotated loadings ===\n")
```

```
##
## === (a)  Unrotated loadings ===
```

```r
print(fa_raw$loadings, cutoff = .30, digits = 3)
```

```
##
## Loadings:
##    PC1    PC2    PC3
## T1  0.536  0.461  0.478
## T2         0.870
## T3  0.514        -0.448
## T4  0.724 -0.366
## T5 -0.416 -0.414  0.649
## T6  0.715         0.420
##
##                  PC1   PC2   PC3
## SS loadings    1.775 1.354 1.073
## Proportion Var 0.296 0.226 0.179
## Cumulative Var 0.296 0.522 0.700
```

I selected 3 factors becuase togeter they reach 70% cumulitive variance which is a common threshold

Factor 1 covers 29.6% of variance and seems to contrast t5 against t 1,3,4,6 as it has positive wieghts for those and a negative weight for t5.. Factor 2 accounts for 22.6% of variance is largely accounting for T2 and has opposite sighns for t 4 and 5.. PC3 accounts for another 17% of variance, and contrasts t3 agaisnt t1,5 and 6. Altogether thses are hard to interpret.

```
communalities  <- fa_raw$communality
specificities  <- 1 - communalities

cat("\nCommunalities:\n");   print(round(communalities, 3))
```

```
##
## Communalities:

##    T1    T2    T3    T4    T5    T6
## 0.729 0.806 0.529 0.670 0.766 0.702
```

```
cat("\nSpecificities:\n");   print(round(specificities, 3))
```

```
##
## Specificities:

##    T1    T2    T3    T4    T5    T6
## 0.271 0.194 0.471 0.330 0.234 0.298
```

The three components accoutn for a lot of the variance. we see that 4 of 6 commonalities are above .7 meaning that 70% of their variance is accounted for. T2 is the most explained, and T3 is the least explained with .529 commonality.

T3 has a high specificity which indicates that the test has unique variance that the selected factors do not capture.

b

```
fa_rot <- factanal(tests, factors = m,
                   rotation = "varimax", scores = "regression")
print(fa_rot, digits = 3, cutoff = 0.30)
```

```
##
## Call:
## factanal(x = tests, factors = m, scores = "regression", rotation = "varimax")
##
## Uniquenesses:
##    T1    T2    T3    T4    T5    T6
## 0.005 0.703 0.866 0.378 0.005 0.726
##
## Loadings:
##    Factor1 Factor2 Factor3
## T1  0.996
```

```
## T2         -0.328  -0.387
## T3                  0.345
## T4                  0.784
## T5          0.977
## T6  0.367           0.366
##
##                Factor1 Factor2 Factor3
## SS loadings      1.178   1.078   1.061
## Proportion Var   0.196   0.180   0.177
## Cumulative Var   0.196   0.376   0.553
##
## The degrees of freedom for the model is 0 and the fit was 0.0165
```

```
obs26_scores <- fa_rot$scores[26, ]
obs26_scores
```

```
##    Factor1    Factor2    Factor3
## -3.5288297 -0.4527419  0.6954494
```

This is now easier to interpret because there are clear factors for certain items.

Factor 1 is clearly accountign fo T1 variance wiht some T6, Factor 2, is focused on T5 with some negative T2 and factor 3 is spread across several. Now, T1, T5 and T4 are only tied to one matrix which makes things simpler.

c

```
scores_all <- fa_rot$scores         # n  × 3 matrix (n = number of students)

obs26 <- scores_all[26, ]           # a named vector
print(obs26, digits = 3)
```

```
## Factor1 Factor2 Factor3
##  -3.529  -0.453   0.695
```

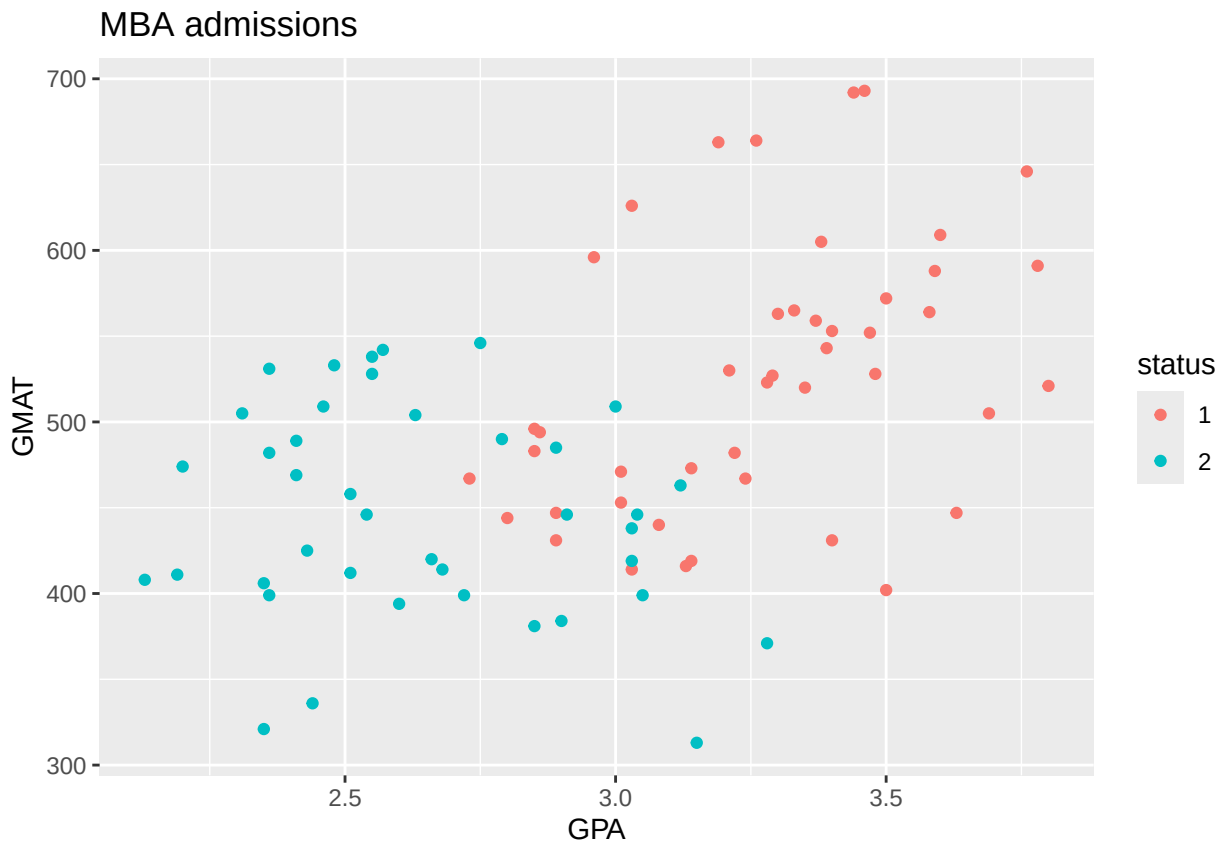
Student 26 is very weak for factor 1 focused on T1 and T6. They are slightly below average for T5 (with some signal from T2), and moderately above average for T4 with some loading from T3 and T6. Altogether it is hard to give a text interpretation here without knowing whwat each test asks.

Question 2

##a

```
load("Data/admission.RData")
colnames(admission) <- c("GPA","GMAT","status")
admission <- as_tibble(admission) %>% mutate(status = factor(status))

g1 <- filter(admission, status == 1)
g2 <- filter(admission, status == 2)
```

```
ggplot(admission, aes(GPA, GMAT, colour = status)) + geom_point() +
  labs(title = "MBA admissions")
```




```
xbar1 <- colMeans(dplyr::select(g1, GPA, GMAT))
xbar2 <- colMeans(dplyr::select(g2, GPA, GMAT))
S1    <- cov(dplyr::select(g1, GPA, GMAT))
S2    <- cov(dplyr::select(g2, GPA, GMAT))
Spooled <- ((nrow(g1)-1)*S1 + (nrow(g2)-1)*S2)/(nrow(g1)+nrow(g2)-2)

cat("\nGroup means:\n")
```

```
##
## Group means:
```

```
print(round(rbind(Admit = xbar1, Deny = xbar2), 2))
```

```
##         GPA   GMAT
## Admit 3.27 526.11
## Deny  2.64 446.08
```

```
cat("\nGroup covariance matrices:\n\nS1 (Admit):\n")
```

```
##
```

```
## Group covariance matrices:
##
## S1 (Admit):
```

```
print(round(S1, 3))
```

```
##        GPA     GMAT
## GPA  0.080    9.216
## GMAT 9.216 6215.374
```

```
cat("\nS2 (Deny):\n")
```

```
##
## S2 (Deny):
```

```
print(round(S2, 3))
```

```
##         GPA     GMAT
## GPA   0.088   -3.268
## GMAT -3.268 3823.763
```

```
cat("\nPooled covariance (Spooled):\n")
```

```
##
## Pooled covariance (Spooled):
```

```
print(round(Spooled, 3))
```

```
##        GPA     GMAT
## GPA  0.084    3.350
## GMAT 3.350 5091.605
```

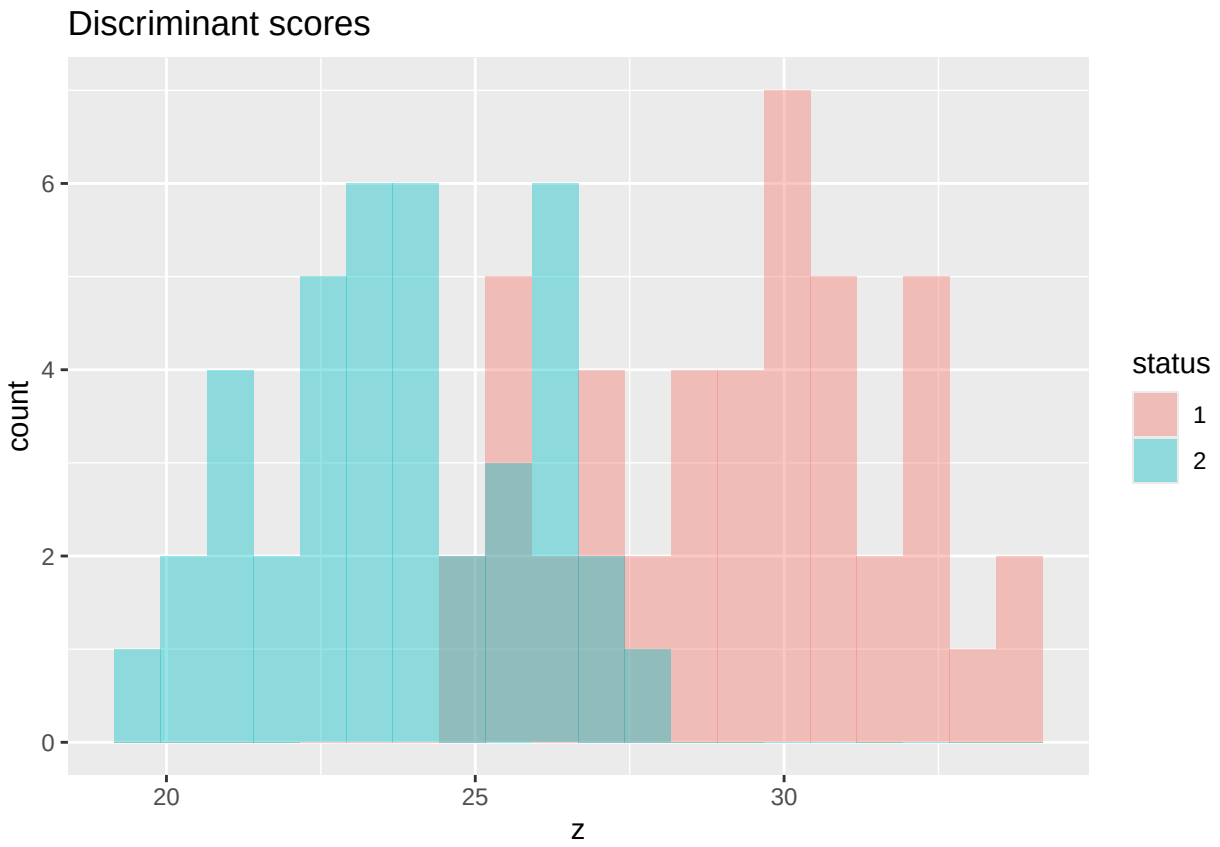
b

```
a <- solve(Spooled, xbar1 - xbar2)
a
```

```
##        GPA       GMAT
## 7.14806424 0.01101639
```

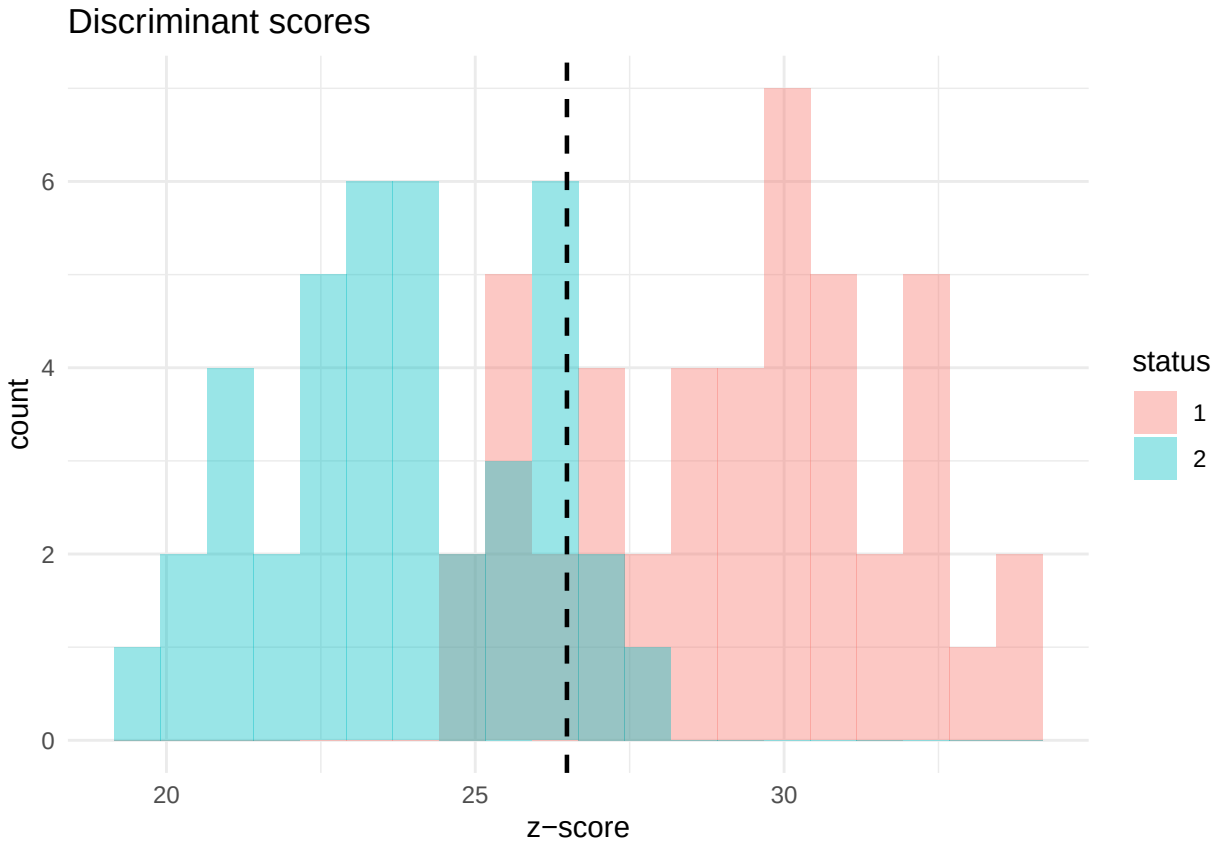
c

```
Z <- as.matrix(dplyr::select(admission, GPA, GMAT)) %*% a   # n×1
admission$z <- drop(Z)

ggplot(admission, aes(x = z, fill = status)) +
  geom_histogram(alpha = 0.4, bins = 20, position = "identity") +
  labs(title = "Discriminant scores")
```



d

```
z_cut <- as.numeric(crossprod(a, (xbar1 + xbar2)/2))  # midpoint

ggplot(admission, aes(x = z, fill = status)) +
  geom_histogram(alpha = 0.4, bins = 20, position = "identity") +
  geom_vline(xintercept = z_cut, linetype = "dashed", linewidth = 0.8) +
  labs(title = "Discriminant scores", x = "z-score") +
  theme_minimal()
```



e

```
## ----- (e)  classify (3, 500) -----
new_pt <- c(3, 500)
z_new  <- crossprod(a, new_pt)
group  <- ifelse(z_new > z_cut, "Admit", "Reject")     # 1 = admit

print(group)
```

```
##      [,1]
## [1,] "Admit"
```

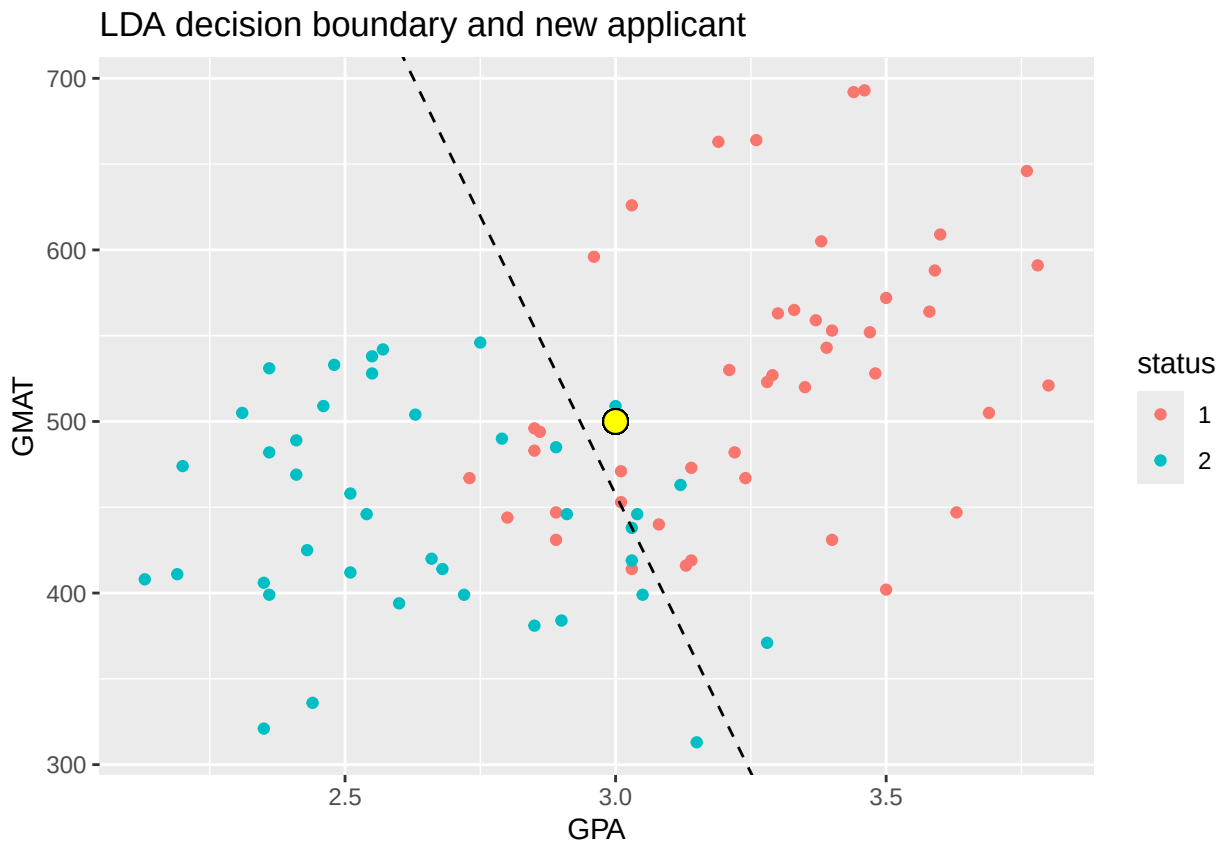
This person should be Admitted.

f

```
# direction vector
slope <- -a[1]/a[2]
intercept <- z_cut/a[2]

ggplot(admission, aes(GPA, GMAT, colour = status)) +
  geom_point() +
  geom_abline(slope = slope, intercept = intercept, lty = 2) +
  geom_point(aes(x = new_pt[1], y = new_pt[2]), colour = "black",
```

```
            shape = 21, size = 4, fill = "yellow") +
  labs(title = "LDA decision boundary and new applicant")
```

```
## Warning in geom_point(aes(x = new_pt[1], y = new_pt[2]), colour = "black", : All aesthetics have leng
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
```



g

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
lda_fit <- lda(status ~ GPA + GMAT, data = admission)
lda_fit$scaling
```

```
##               LD1
## GPA  -3.071443853
## GMAT -0.004733621
```

```
cat("scaling", a/lda_fit$scaling)
```

```
## scaling -2.327265 -2.327265
```

This appears to be a scaled and flipped version of my coefficients. specifically, my coefficients are equal to -2.327 times the LDA library coefficients. This is because of the way variance scaling is maanged in LDA.