Log

Joshua Tolhuis

9/14/2021

EDA of Urinary Biomarkers for cancer

Assignment Introduction:

insert text here

Data

The data gained was from John Davis, uploaded to kaggle John Davis. He had gotten his data from a paper with the title "A combination of urinary biomarker panel and PancRisk score for earlier detection of pancreatic caner: A case-control study", this is the link to the paper.

setup and load code book:

```
codebook <- read.csv("Debernardi et al 2020 documentation.csv", sep = ",", header = T)

knitr::kable(codebook)
```

Column.name	Original.column.name	Details
sample_id	Sample ID	Unique string identifying each subject
patient_cohort	Patient's Cohort	Cohort 1, previously used samples; Cohort 2, newly added samples
sample_origin	Sample Origin	BPTB: Barts Pancreas Tissue Bank, London, UK; ESP: Spanish National Cancer Research Centre, Madrid, Spain; LIV: Liverpool University, UK; UCL: University College London, UK
age	Age	Age in years
sex	Sex	M = male, F = female
diagnosis	Diagnosis (1=Control, 2=Benign, 3=PDAC)	1 = control (no pancreatic disease), 2 = benign hepatobiliary disease (119 of which are chronic pancreatitis); 3 = Pancreatic ductal adenocarcinoma, i.e. pancreatic cancer
stage	Stage	For those with pancratic cancer, what stage was it? One of IA, IB, IIA, IIIB, III, IV
benign_sample_diagnosis	Benign Samples Diagnosis	For those with a benign, non-cancerous diagnosis, what was the diagnosis?
plasma_CA19_9	Plasma CA19-9 U/ml	Blood plasma levels of CA 19–9 monoclonal antibody that is often elevated in patients with pancreatic cancer. Only assessed in 350 patients (one goal of the study was to compare various CA 19-9 cutpoints from a blood sample to the model developed using urinary samples).
creatinine	Creatinine mg/ml	Urinary biomarker of kidney function
LYVE1	LYVE1 ng/ml	Urinary levels of Lymphatic vessel endothelial hyaluronan receptor 1, a protein that may play a role in tumor metastasis

Column.name	Original.column.name	Details
REG1B	REG1B ng/ml	Urinary levels of a protein that may be associated with pancreas regeneration.
TFF1	TFF1 ng/ml	Urinary levels of Trefoil Factor 1, which may be related to regeneration and repair of the urinary tract
REG1A	REG1A ng/ml	Urinary levels of a protein that may be associated with pancreas regeneration. Only assessed in 306 patients (one goal of the study was to assess REG1B vs REG1A)

Here is the corresponding data file and it's first 6 entries:

```
data <- read.csv("Debernardi et al 2020 data.csv", header = T, sep = ",")
knitr::kable(head(data))
```

sample_id	patient_cohort	sample_origin	age	sex	diagnosis	stage	benign_sample_diagnosis	plasma_CA19_9	creatinine	LYVE1	REG1B	TFF1	REG1A
S1	Cohort1	BPTB	33	F	1			11.7	1.83222	0.893219	52.94884	654.2822	1262.000
S10	Cohort1	BPTB	81	F	1			NA	0.97266	2.037585	94.46703	209.4882	228.407
S100	Cohort2	BPTB	51	M	1			7.0	0.78039	0.145589	102.36600	461.1410	NA
S101	Cohort2	BPTB	61	M	1			8.0	0.70122	0.002804	60.57900	142.9500	NA
S102	Cohort2	BPTB	62	M	1			9.0	0.21489	0.000859	65.54000	41.0880	NA
S103	Cohort2	BPTB	53	M	1			NA	0.84825	0.003393	62.12600	59.7930	NA

Loading up libraries to be used in the EDA and log

```
if(!require(devtools)) install.packages("devtools")
```

```
## Loading required package: devtools
```

```
## Loading required package: usethis
```

```
devtools::install_github("sinhrks/ggfortify")
```

```
## Skipping install of 'ggfortify' from a github remote, the SHA1 (195b1fb1) has not changed since last
##   Use `force = TRUE` to force installation
```

```
devtools::install_github("AckerDWM/gg3D")
```

```
## Skipping install of 'gg3D' from a github remote, the SHA1 (ffdd837d) has not changed since last insta
##   Use `force = TRUE` to force installation
```

```
library("gg3D")
```

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

```
## Warning in fun(libname, pkgname): couldn't connect to display ":0"
```

```
library(ggplot2)
library(tidyr)
library(gridExtra)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

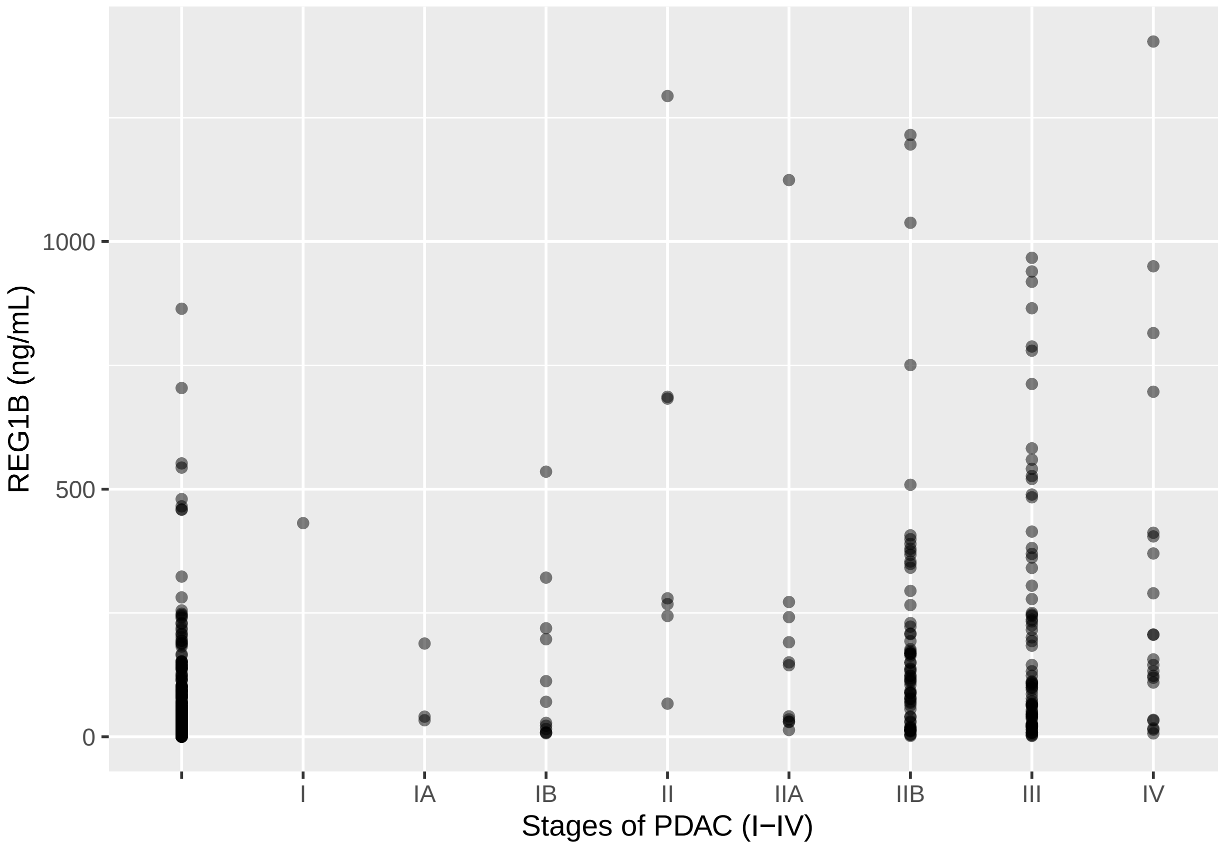
```r
library(cluster)
library(ggfortify)
```

Intro

At first, I wanted to know if REG1B could be used only to predict PDAC, but after reconsideration I want to find out what the minimum required data is to find out if a patient has PDAC. Even so i first wanted to run some tests on the newly improved biomarker REG1B to find out it's impact, and it's change relative to REG1A.

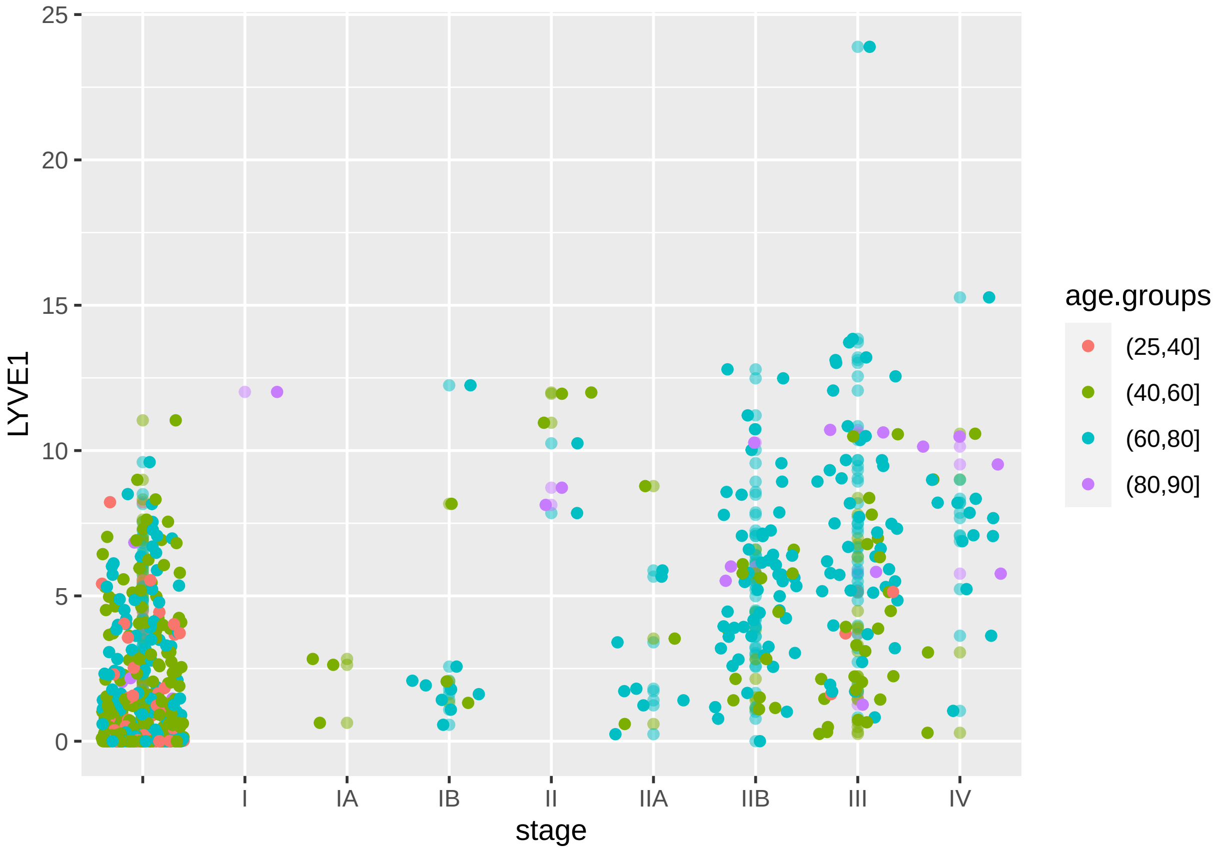
Let's take a first look at the effect of REG1B stand alone, what conclusions can be made by looking at the markers found at different stages.

```r
ggplot(data = data, mapping = aes(x = stage, y = REG1B)) +
        geom_point(alpha = 0.5) +
        xlab("Stages of PDAC (I-IV)") +
        ylab("REG1B (ng/mL)")
```



In this plot becomes clear, that REG1B is has teh same values at the stages "0, IIB, III and IV" relative to the others. This rises the question with which other data values the results be improved so that stage 0 and I - IIA can be recognized more easily. Lets first try finding patterns using only bodily data such as, age, sex, and body fluids or hormones.

```
age.groups <- cut(data$age, breaks = c(25 ,40, 60, 80, 90))
ggplot(data = data, mapping = aes(x = stage, y = LYVE1, col = age.groups)) +
        geom_point(alpha = 0.5) +
        geom_jitter()
```



in this plot the REG1B has gained a third dimension, this dimension is age. I wanted to figure out if age might have an influence on the output of the bio marker. it looks like this however is not the case, age is pretty randomly diveded except for people under 40.

```
sum(data$age < 40)
```

```
## [1] 43
```

```
sum(data$age < 40 & data$stage == "")
```

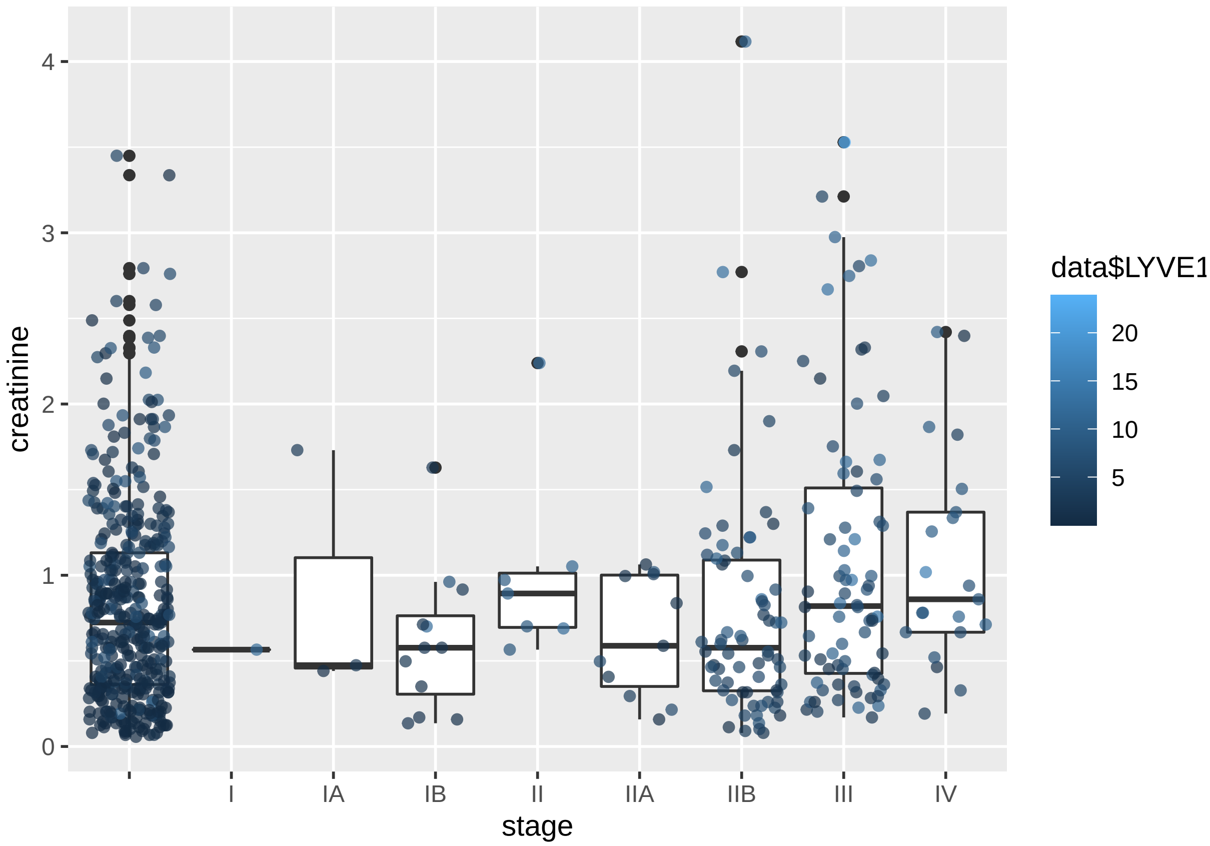
```
## [1] 40
```

But I found out that there were only 43 people under 40 in this data set and only 3 of them had a state assigned to them. So I concluded, age had no effect on the effects of this biomarker.

I also tried another few values as 3rd dimension to see if any would be promising:

```
creatinine.groups <- cut(data$creatinine, breaks = c(0.05654 ,1.071623 ,2.086695, 3.101768, 4.11685))
ggplot(data = data, mapping = aes(x = stage, y = creatinine)) +
        geom_boxplot() +
        geom_jitter(alpha = 0.7, aes(col = data$LYVE1))
```

```
## Warning: Use of `data$LYVE1` is discouraged. Use `LYVE1` instead.
```



scaling function:

```
scale_min_max <- function(x) {
     (x - min(x)) / (max(x) - min(x))
}
```

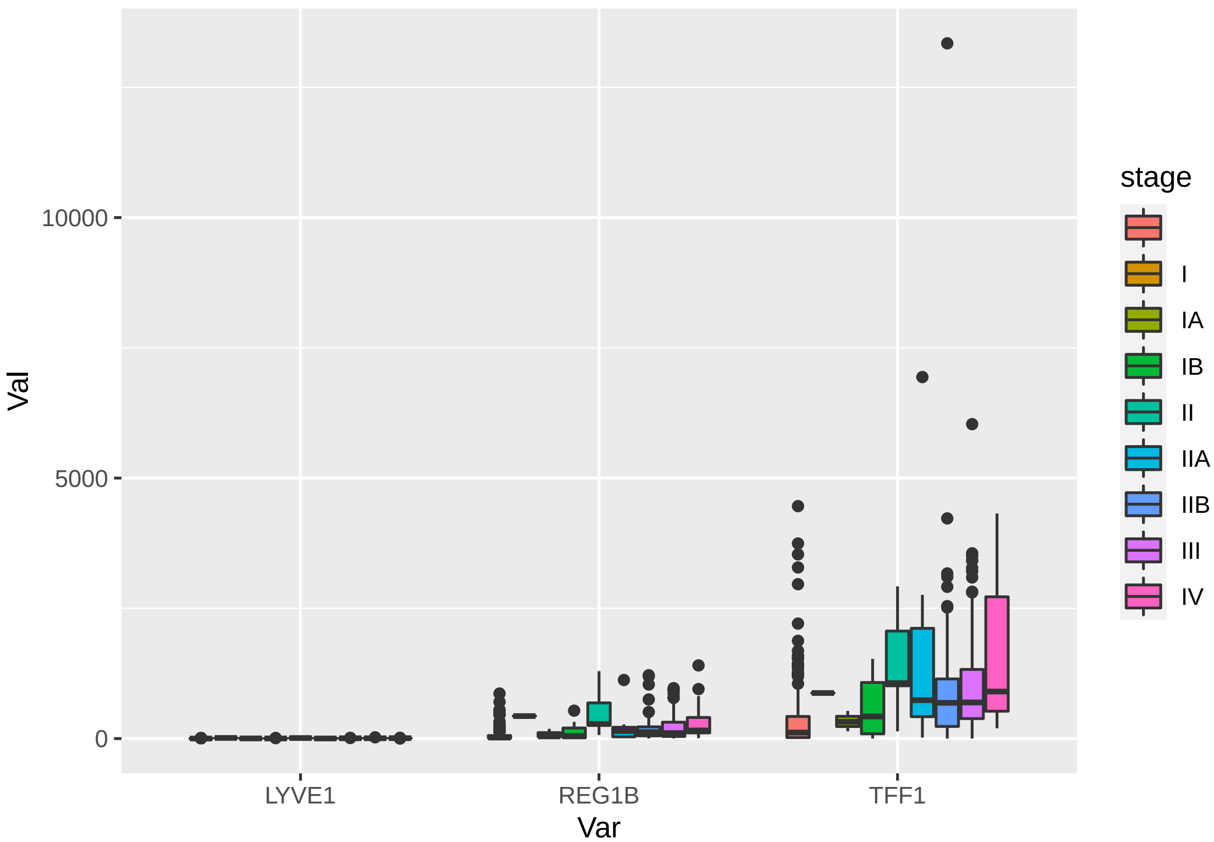
outlier removal function:

```
remove_outliers <- function(x, na.rm = TRUE, ...) {
  qnt <- quantile(x, probs=c(.25, .75), na.rm = na.rm, ...)
  H <- 1.5 * IQR(x, na.rm = na.rm)
  y <- x
  y[x < (qnt[1] - H)] <- NA
  y[x > (qnt[2] + H)] <- NA
  y
}
```

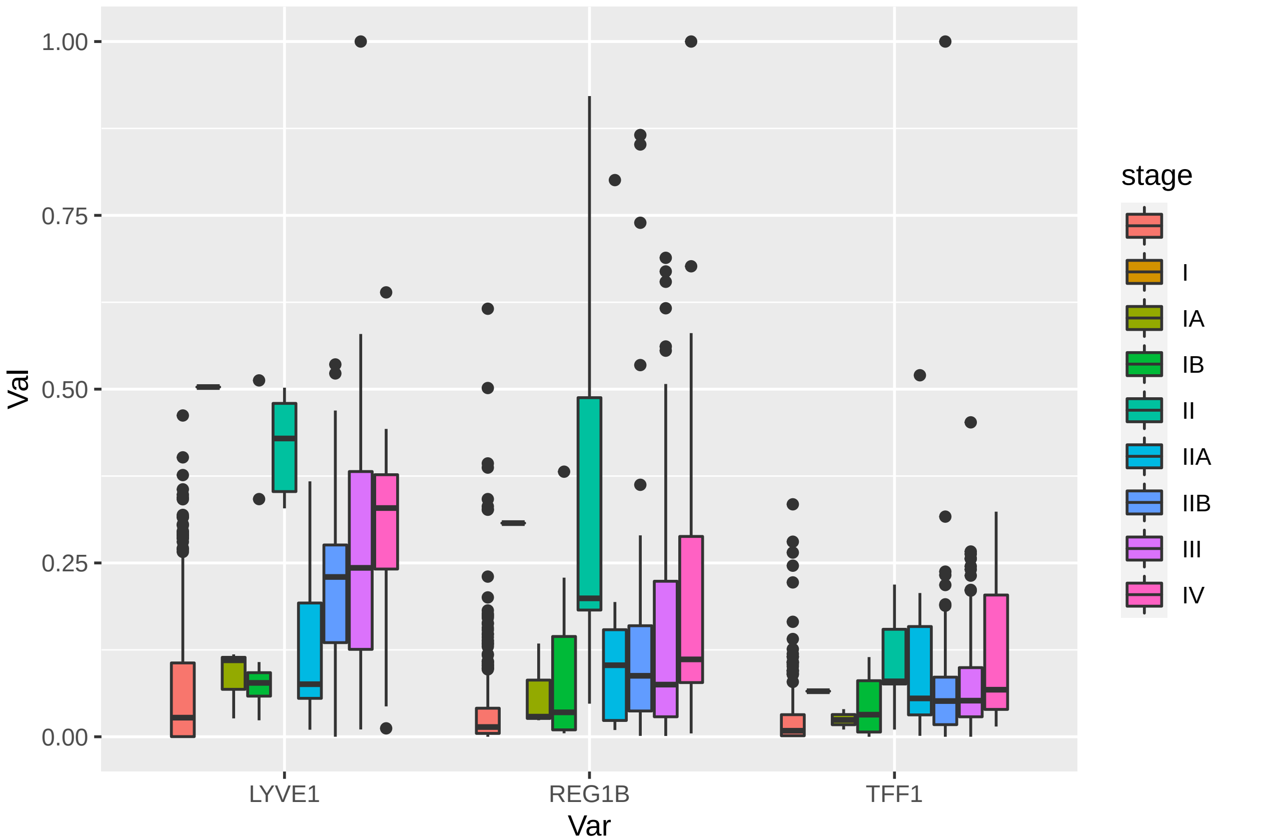
several boxplots on the biomarkers, first with normal data, second with scaled data, third with scaled data and outliers removed

```
scaled_data <- data

scaled_data$REG1B <- scale_min_max(scaled_data$REG1B)
scaled_data$TFF1 <- scale_min_max(scaled_data$TFF1)
scaled_data$LYVE1 <- scale_min_max(scaled_data$LYVE1)

p1 <- pivot_longer(data = data, cols = c(REG1B,TFF1,LYVE1), names_to = "Var", values_to = "Val") %>%
        ggplot(aes(x = Var, y = Val, fill = stage)) +
        geom_boxplot()
```

```r
p2 <- pivot_longer(data = scaled_data, cols = c(REG1B,TFF1,LYVE1), names_to = "Var", values_to = "Val")
        ggplot(aes(x = Var, y = Val, fill = stage)) +
        geom_boxplot()
p3 <- pivot_longer(data = scaled_data, cols = c(REG1B,TFF1,LYVE1), names_to = "Var", values_to = "Val")
        ggplot(aes(x = Var, y = remove_outliers(Val), fill = stage)) +
        geom_boxplot()
p1
```

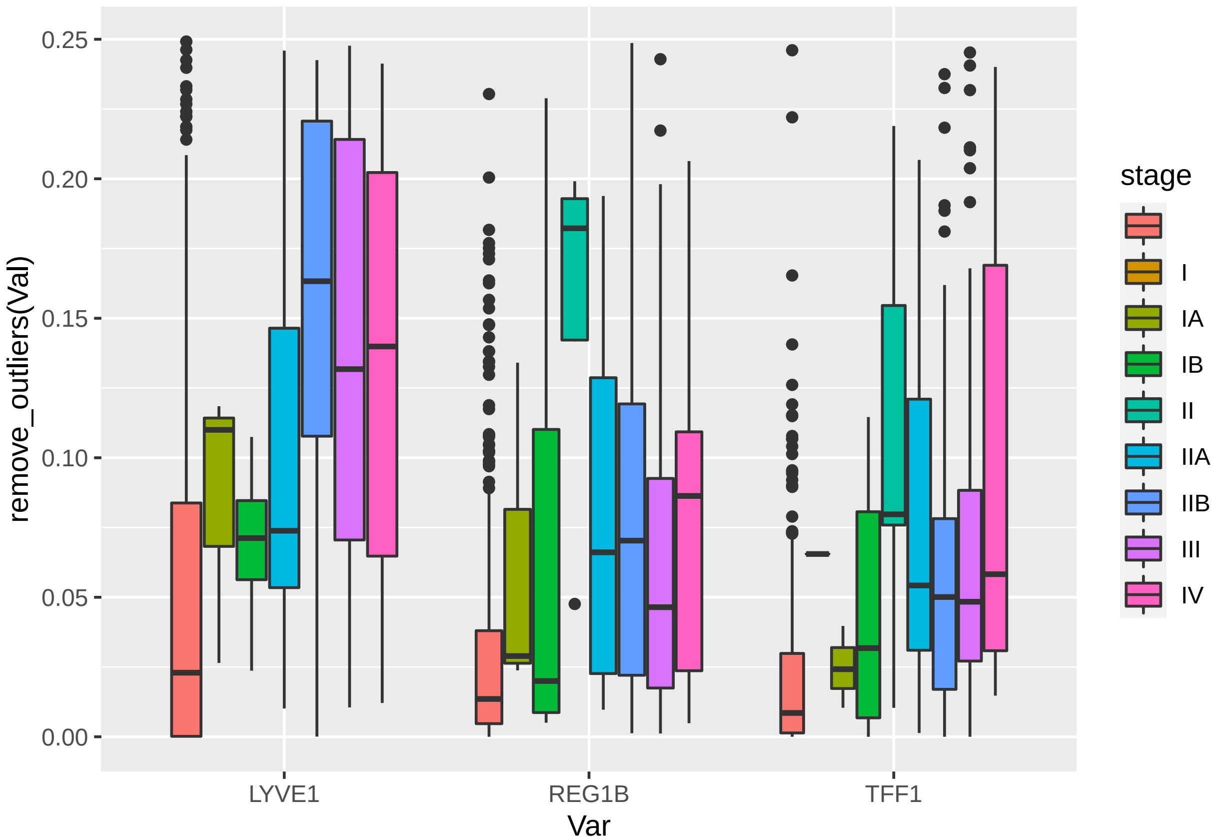



```r
p2
```

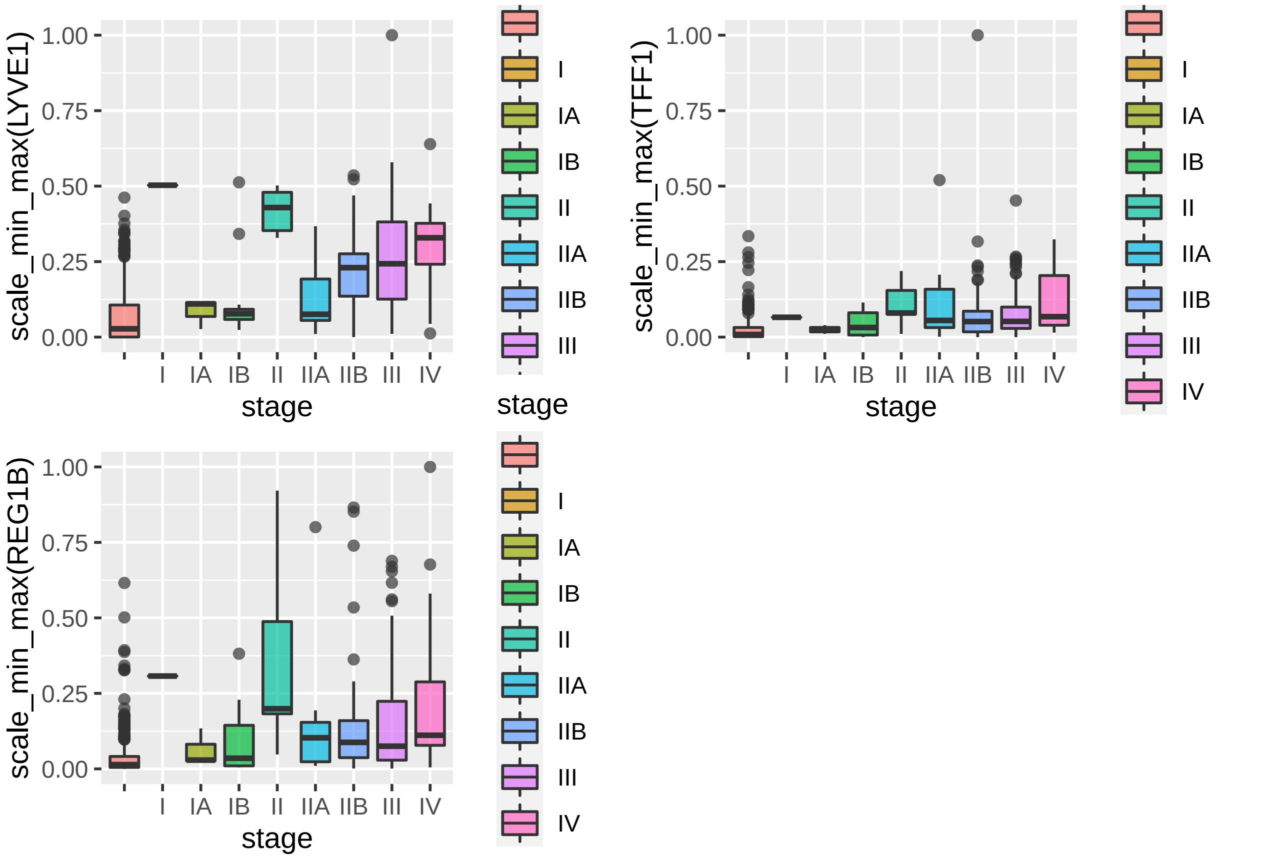



```
p3
```

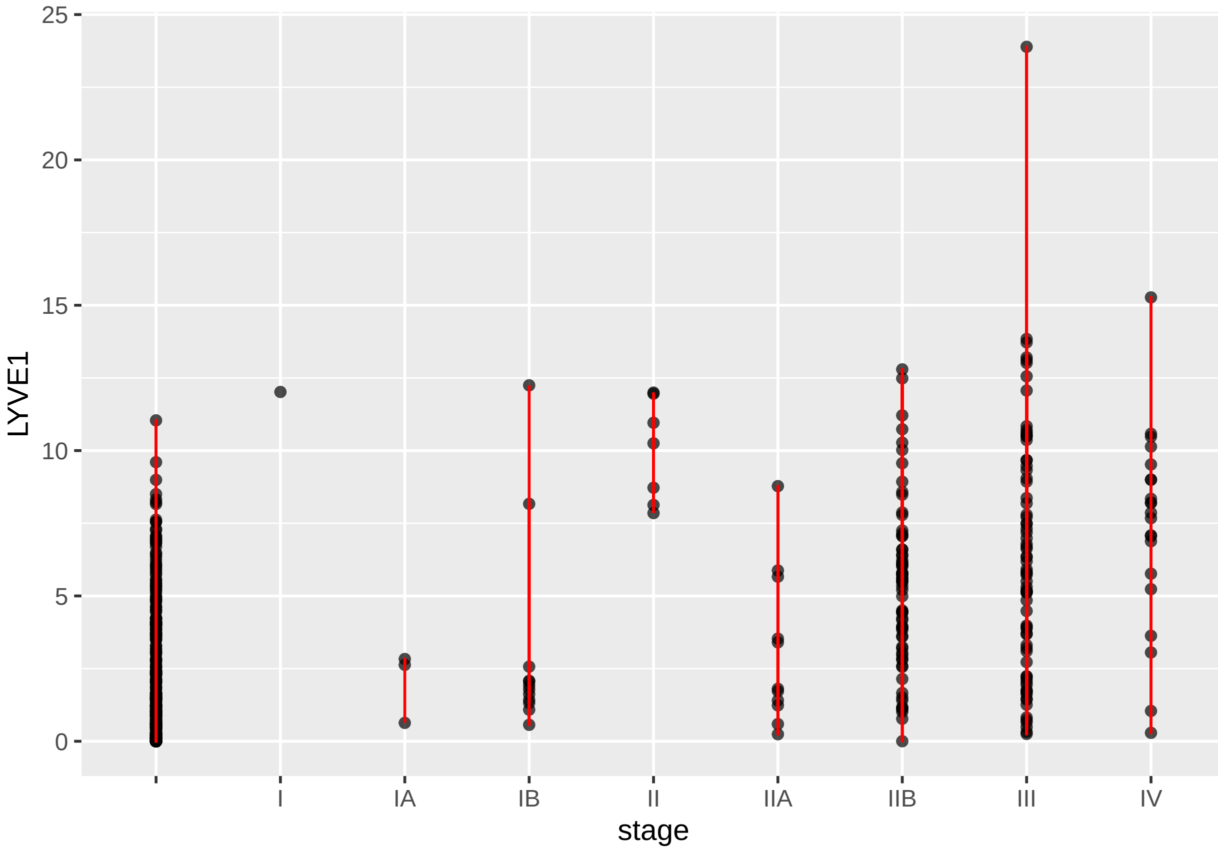
```
## Warning: Removed 179 rows containing non-finite values (stat_boxplot).
```




```r
p1 <- ggplot(data = data, mapping = aes(x = stage, y = scale_min_max(LYVE1), fill = stage)) +
        geom_boxplot(alpha = 0.7)

p2 <- ggplot(data = data, mapping = aes(x = stage, y = scale_min_max(TFF1), fill = stage)) +
        geom_boxplot(alpha = 0.7)

p3 <- ggplot(data = data, mapping = aes(x = stage, y = scale_min_max(REG1B), fill = stage)) +
        geom_boxplot(alpha = 0.7)

grid.arrange(p1, p2, p3, nrow = 2)
```




```r
ggplot(data = data, mapping = aes(x = stage, y = LYVE1)) +
        geom_point(alpha = 0.7) +
        geom_line(color = "red", aes())
```




```
#ggplot(data = data, mapping = aes(x = stage, y = scale_min_max(TFF1), fill = stage)) +
#         geom_boxplot(alpha = 0.7)

#ggplot(data = data, mapping = aes(x = stage, y = scale_min_max(REG1B), fill = stage)) +
#         geom_boxplot(alpha = 0.7)
```

```
df <- as.data.frame(data[392:590,])
row.names(df) <- paste(df$stage, row.names(df), sep="_")

df[11:13]
```

```
##                LYVE1      REG1B         TFF1
## I_392    12.01715000 431.422530 8.740997e+02
## IA_393    2.62842500  40.620818 5.299840e+02
## IA_394    2.83054100  33.406150 3.231758e+02
## IA_395    0.63243260 188.253000 1.386300e+02
## IB_396   12.24582000 196.921830 1.529183e+03
## IB_397    2.56734600  15.695743 2.970097e+02
## IB_398    1.42287700   7.085142 4.499118e-02
## IB_399    2.08153500  22.244200 5.480474e+01
## IB_400    8.16755400 535.281600 1.099519e+03
## IB_401    1.78321200 112.348000 4.785100e+01
## IB_402    1.61933900   8.675667 6.108991e+02
## IB_403    1.92032700 219.129024 4.146139e+02
## IB_404    0.56583980 321.308160 1.068504e+03
## IB_405    1.08506400  70.605010 1.115292e+03
## IB_406    1.31914400  28.068943 1.024494e+02
```

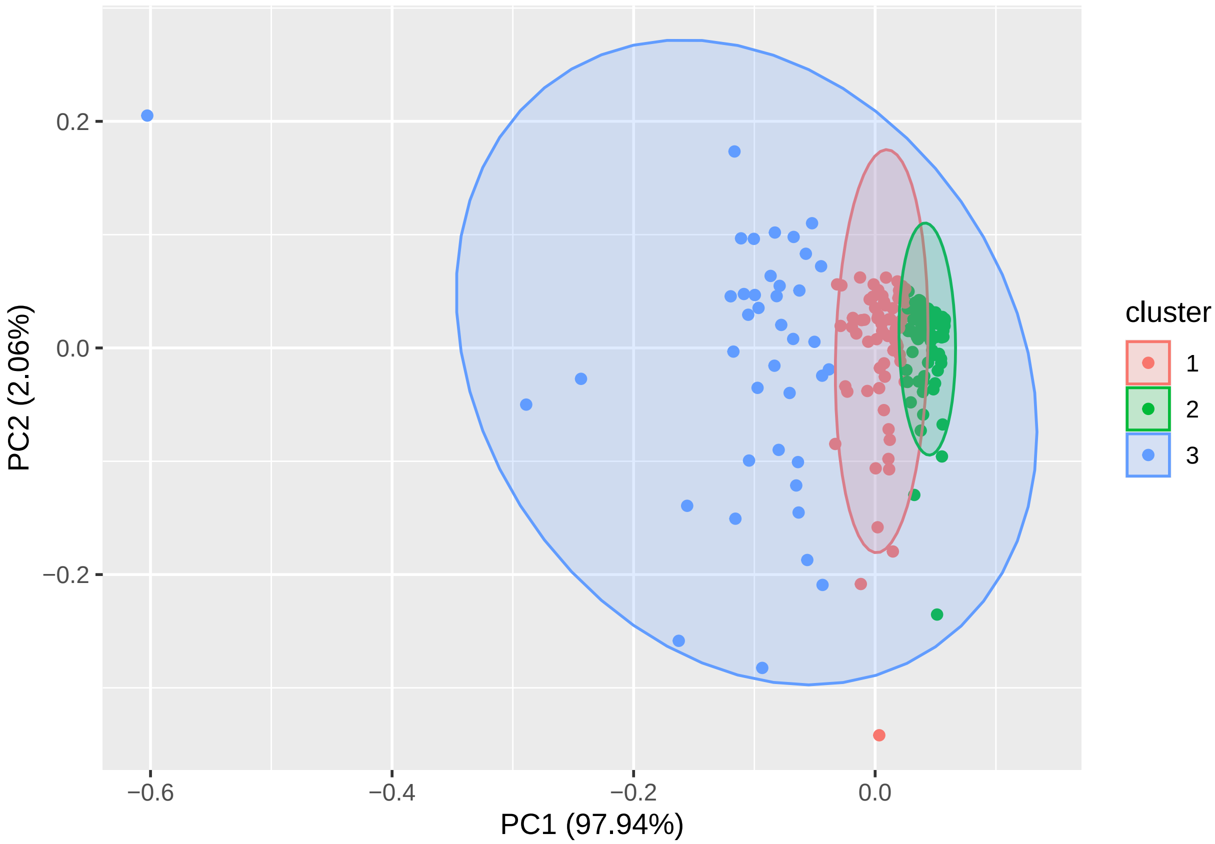
```
## IB_407   2.05643800    8.432788 4.335330e+02
## II_408   7.84910500  267.856820 1.063949e+03
## II_409  11.95479000  682.898790 1.047710e+03
## II_410   8.72221200  243.906740 1.730613e+03
## II_411  10.95695000 1293.819450 2.921507e+03
## II_412  10.25015000  279.524840 9.778096e+02
## II_413   8.12974700   66.833074 1.383245e+02
## II_414  11.99636000  686.596400 2.395081e+03
## IIA_415  5.87575900  272.131400 2.759204e+03
## IIA_416  5.66198400   30.754640 3.967304e+02
## IIA_417  1.72242900   41.140085 4.079489e+02
## IIA_418  1.23380200  144.475310 1.803015e+03
## IIA_419  1.40685700   13.618654 4.317260e+02
## IIA_420  8.77698200  150.187310 1.050400e+03
## IIA_421  0.59247880   29.716092 7.109413e+02
## IIA_422  3.40208400  241.579240 2.428988e+03
## IIA_423  3.53085500 1124.108000 6.939098e+03
## IIA_424  0.24221000   34.981375 1.785104e+01
## IIA_425  1.80642200  190.812000 7.366730e+02
## IIB_426  2.55849100   15.176469 1.013080e+02
## IIB_427  8.57669400  115.915310 2.542312e+03
## IIB_428  7.78674100  222.366130 1.541649e+03
## IIB_429 12.48489000  349.068580 3.169306e+03
## IIB_430  2.58967400   88.393900 2.240890e+02
## IIB_431  4.16958200   14.657195 2.870377e+02
## IIB_432  7.87098500  341.279540 2.160880e+03
## IIB_433  4.50149500  135.647680 1.106790e+03
## IIB_434  3.82963300   92.548050 3.995643e+02
## IIB_435  3.95178900   40.101544 6.983854e+02
## IIB_436  6.59851900   89.432420 3.456236e+02
## IIB_437  5.73324200  265.985020 2.913086e+03
## IIB_438  3.90089100  207.307240 1.642566e+02
## IIB_439  1.40685700  294.545020 3.269259e+02
## IIB_440  8.92967800  374.512950 3.103221e+03
## IIB_441  6.22186900   63.468804 5.831935e+02
## IIB_442  4.99012200   88.393900 1.014292e+03
## IIB_443  5.33623300  229.635980 1.466508e+03
## IIB_444  8.48176900  170.438940 1.014292e+03
## IIB_445  3.59549900  379.705690 2.516135e+03
## IIB_446  6.41528400   68.661523 5.723529e+02
## IIB_447  6.58834000  132.532050 8.113080e+02
## IIB_448  1.10146500   20.369188 4.014759e+02
## IIB_449  1.66135000   12.580106 3.843465e+02
## IIB_450  5.73324200  167.323310 1.037416e+03
## IIB_451  5.79432100  398.918730 1.597547e+03
## IIB_452  6.38474500   76.450640 1.045124e+03
## IIB_453  4.42005700  173.312720 7.740403e+02
## IIB_454  5.51946800  168.013872 7.901210e+02
## IIB_455  6.08953300  353.881280 1.263524e+03
## IIB_456  3.61585800   13.563616 5.786800e+00
## IIB_457  5.62126600  406.730640 8.977089e+02
## IIB_458  3.24938800   18.368096 1.394943e+02
## IIB_459  7.05660700  389.514560 1.632247e+03
## IIB_460  7.06678600   31.580432 2.117685e+02
```

```
## IIB_461  3.03561400  168.107840 9.411622e+02
## IIB_462  5.47874900  104.848800 5.469793e+02
## IIB_463  7.25002200  508.825920 8.746521e+02
## IIB_464  2.81166000  176.916080 1.337829e+03
## IIB_465  3.93143000   11.962120 2.383725e+02
## IIB_466  1.50568000    3.859414 2.470104e-02
## IIB_467 10.02264000  367.560200 9.820685e+02
## IIB_468  6.14269200   76.438300 6.729489e+02
## IIB_469  0.00126672    1.769536 2.470104e-02
## IIB_470  2.14329200  118.504400 1.710781e+02
## IIB_471  1.01339200  123.010000 9.294080e+02
## IIB_472  4.23017700   55.880000 4.523300e+01
## IIB_473  5.60824400  137.768470 1.730682e+03
## IIB_474  9.56575000  193.089000 8.440440e+02
## IIB_475 11.21034000 1215.168000 9.510850e+02
## IIB_476  0.77350730  149.629000 9.827200e+01
## IIB_477  7.14079100   40.543000 1.960620e+02
## IIB_478  5.20792800   80.119575 6.635936e+02
## IIB_479  1.14499000    4.729000 1.554450e+02
## IIB_480 12.79400000 1037.972000 4.225523e+03
## IIB_481  6.06330900  111.083000 4.374600e+02
## IIB_482  5.50488600   17.621000 1.670490e+02
## IIB_483  4.45214600   13.404489 8.422872e+00
## IIB_484  2.95921000   29.138000 2.096760e+02
## IIB_485  5.77253700  750.559000 2.416474e+03
## IIB_486  4.45831300  163.837170 8.191460e+02
## IIB_487  3.19371400   10.755955 2.796968e+02
## IIB_488 10.73328000 1195.972000 1.334430e+04
## IIB_489  2.82879800  208.371000 3.338520e+02
## IIB_490  6.01137600  150.056000 2.631630e+02
## IIB_491 10.27304000  113.094000 1.256712e+03
## IIB_492  1.17214100  123.270000 4.353700e+01
## IIB_493  5.77897500   71.054000 2.356180e+02
## III_494 10.36449000   64.606514 3.624514e+02
## III_495 13.72179000  582.435750 3.092892e+03
## III_496  2.22587900   24.173807 3.558191e+02
## III_497  5.50003100    7.644714 4.123118e+02
## III_498  9.04443100  215.752320 6.201404e+02
## III_499 23.89032300   23.156637 3.555723e+03
## III_500  0.31642770   65.794463 1.194496e+03
## III_501 10.56197000    8.407595 7.174239e+02
## III_502  7.71398100   99.444730 1.203845e+03
## III_503 13.01499000  414.261680 3.272766e+03
## III_504  7.49433500   77.321230 1.423229e+03
## III_505  5.30569400   40.957196 2.240657e+03
## III_506  2.13979800  236.509070 3.375213e+02
## III_507 10.49736000  108.271240 4.561572e+02
## III_508  5.78414100   36.125614 7.614703e+02
## III_509  4.84760600  483.539420 8.903848e+02
## III_510  5.13263800   54.163914 6.186736e+02
## III_511 13.11355000  967.171250 3.417822e+03
## III_512  9.67279800  369.256580 2.719353e+03
## III_513 12.06503000  939.619750 6.035157e+03
## III_514  1.45775600   62.254731 3.939007e+02
```

```
## III_515  1.71224900   20.536467 2.327158e+02
## III_516  5.82486000   19.087936 3.930478e+02
## III_517  3.09669200  106.819300 6.054024e+02
## III_518  3.98232900   71.218140 5.235307e+02
## III_519  4.48113500   94.613190 9.934400e+02
## III_520  1.62063200   13.747762 3.017950e+02
## III_521  9.46920300   21.630868 9.908814e+02
## III_522  3.70747600   99.190490 7.222401e+02
## III_523  6.35831400  788.087250 3.210585e+03
## III_524  8.18655700   86.221520 8.695662e+02
## III_525  7.31110000   42.027692 6.190477e+02
## III_526  5.72306300   62.317864 3.589346e+02
## III_527 13.84041600  132.248690 2.075596e+03
## III_528 10.71113000  225.065890 1.266132e+03
## III_529  6.98534900   26.716746 7.699986e+02
## III_530  3.30653700  249.902940 6.458575e+02
## III_531 13.20467000  526.529850 2.557174e+03
## III_532  5.18453400   25.633080 7.477938e+02
## III_533  0.25179460  340.948000 3.616364e+02
## III_534 12.55407000  489.043475 2.819214e+03
## III_535  5.16087600  103.856980 8.290471e+01
## III_536  6.19000900  779.647000 2.856528e+01
## III_537  0.81959190   38.536000 4.320099e+02
## III_538  6.62768600  244.176200 5.986167e+02
## III_539 10.83885000  520.583225 4.473889e+02
## III_540  1.25726900   19.988054 2.470104e-02
## III_541  9.32507200  246.670200 9.949241e+02
## III_542  6.68496200   47.699981 7.784544e+02
## III_543 10.48922000  865.288200 1.308224e+03
## III_544 10.62434000  712.375860 7.851042e+02
## III_545  8.36881200  123.235000 1.163021e+03
## III_546  7.79713400  145.016620 1.118769e+03
## III_547  7.18388100   44.535232 2.623092e+02
## III_548  8.93436300  541.053450 3.507253e+03
## III_549  3.93443100   66.462480 3.900155e+02
## III_550  0.73352710   45.244995 6.687273e+02
## III_551  3.68223700  361.897760 2.117232e-02
## III_552  7.47996700  200.640300 1.127803e+03
## III_553  0.65212100    1.651784 3.997348e+01
## III_554  1.70677500    2.632109 2.509692e+01
## III_555  2.23690900  305.035680 2.805295e+03
## III_556  3.87258200  278.070250 2.734758e-02
## III_557  1.43423000   50.632904 1.527972e+01
## III_558  1.76472900   15.285000 5.056000e+01
## III_559  5.13721800  918.738030 2.207570e+03
## III_560  6.78146400  110.141990 4.345447e+02
## III_561  3.19998700  193.186000 3.215290e+02
## III_562  2.72438000  183.879000 1.383490e+03
## III_563  1.94335800    4.557689 3.713591e+01
## III_564  9.66776800  559.547520 1.780555e+03
## III_565  5.91793900  381.221725 1.911565e+03
## III_566  6.33195800   22.762320 4.957253e+02
## III_567  5.10957500  232.920875 3.418001e+03
## III_568  0.48771820  111.710320 6.058256e+02
```

```
## III_569  2.03758500    8.754893 6.655326e+02
## IV_570   9.00533800  144.985040 2.856123e+03
## IV_571   3.05529400   32.890960 1.967100e+02
## IV_572  10.13581000  370.105400 3.351345e+03
## IV_573   6.88245100  109.417560 1.978939e+02
## IV_574   9.52256000  404.495350 3.570915e+03
## IV_575  10.47882000  696.716160 2.720544e+03
## IV_576   3.62908700  119.243250 9.031260e+02
## IV_577   8.99246000 1403.897600 4.320489e+03
## IV_578   7.85949900   34.189659 3.735307e+02
## IV_579  15.27052000  132.879880 2.333089e+03
## IV_580   1.04434500   14.364360 3.273971e+02
## IV_581  10.57861000  206.526460 3.203940e+03
## IV_582   5.23252700  123.104730 7.558209e+02
## IV_583   0.28950000    6.824000 7.992560e+02
## IV_584   5.76434900  950.080000 1.945172e+03
## IV_585   7.08563700  815.136000 2.367590e+03
## IV_586   7.05820900  156.241000 5.251780e+02
## IV_587   8.34120700   16.915000 2.459470e+02
## IV_588   7.67470700  289.701000 5.372860e+02
## IV_589   8.20677700  205.930000 7.225230e+02
## IV_590   8.20095800  411.938275 2.021321e+03
```

```
autoplot(pam(df[c(8,11:13)], 3), frame = TRUE, frame.type = 'norm')
```




```
ggplot(df, aes(x= scale_min_max(TFF1), y = scale_min_max(LYVE1), z = scale_min_max(REG1B), color= stage
  theme_void() +
  axes_3D() +
```

```
  stat_3D() +
        xlab("TFF1")
```