

# unintegrated\_visualisation

Siwanart Ma

2025-03-20

```
library(Seurat)

## Loading required package: SeuratObject

## Loading required package: sp

## 'SeuratObject' was built under R 4.4.0 but the current version is
## 4.4.1; it is recommended that you reinstall 'SeuratObject' as the ABI
## for R may have changed

## 'SeuratObject' was built with package 'Matrix' 1.7.0 but the current
## version is 1.7.3; it is recommended that you reinstall 'SeuratObject' as
## the ABI for 'Matrix' may have changed

##
## Attaching package: 'SeuratObject'

## The following objects are masked from 'package:base':
## 
##     intersect, t

library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr     1.1.4      v readr     2.1.5
## vforcats   1.0.0      v stringr   1.5.1
## v ggplot2   3.5.1      v tibble    3.2.1
## v lubridate 1.9.4      v tidyrr    1.3.1
## v purrr    1.0.4

## -- Conflicts ----- tidyverse_conflicts() --
## 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

library(patchwork)
```

## Load the 3 Seurat objects

```

# Increase the memory
mem.maxVSize(vsize = 60000)

## [1] 60000

# Load 4-6 weeks Seurat object
load("/Users/mayongzhi/Desktop/researchProject/integration/originals/human4_6W.RData")
ls()

## [1] "human4_6W"

# Load 7-11 weeks Seurat object
load("/Users/mayongzhi/Desktop/researchProject/integration/originals/human7_11W.RData")
ls()

## [1] "human4_6W" "seurat"

human7_11W <- seurat
rm(seurat)

# Load 12-20 weeks Seurat object
load("/Users/mayongzhi/Desktop/researchProject/integration/originals/human12_20W.RData")
ls()

## [1] "human4_6W"                 "human7_11W"
## [3] "humanpancreas.combined.sct"

human12_20W <- humanpancreas.combined.sct
rm(humanpancreas.combined.sct)

```

## Visualisation

### human4\_6W

The Seurat object is filtered by nCount\_RNA > 4000 & nCount\_RNA < 50000 & nFeature\_RNA > 500 & nFeature\_RNA < 8000 & percent.mt < 15

Hence, we don't do filtering again.

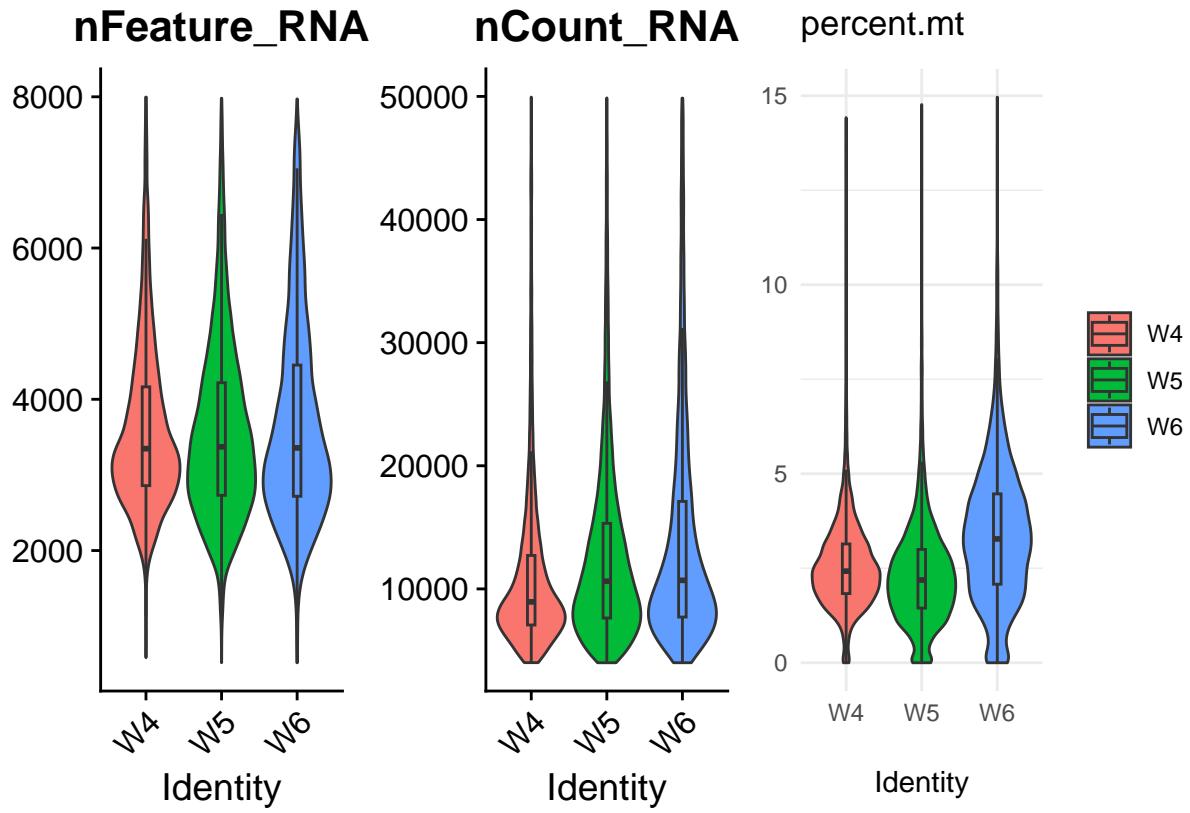
```

p <- VlnPlot(human4_6W, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"),
              ncol = 3, group.by = "days", pt.size = 0)

# Apply boxplot to each plot separately
p <- lapply(p, function(x) x + geom_boxplot(width = 0.1, outlier.shape = NA))

# Print the updated plots
patchwork::wrap_plots(p, ncol = 3) + theme_minimal()

```



Add a column “time” to mark object source for integration

```
human4_6W$time <- "4_6"
```

## human7\_11W

The Seurat object is filtered by  $nCount\_RNA > 4000 \& nCount\_RNA < 50000 \& nFeature\_RNA > 500 \& nFeature\_RNA < 8000 \& percent.mt < 15$

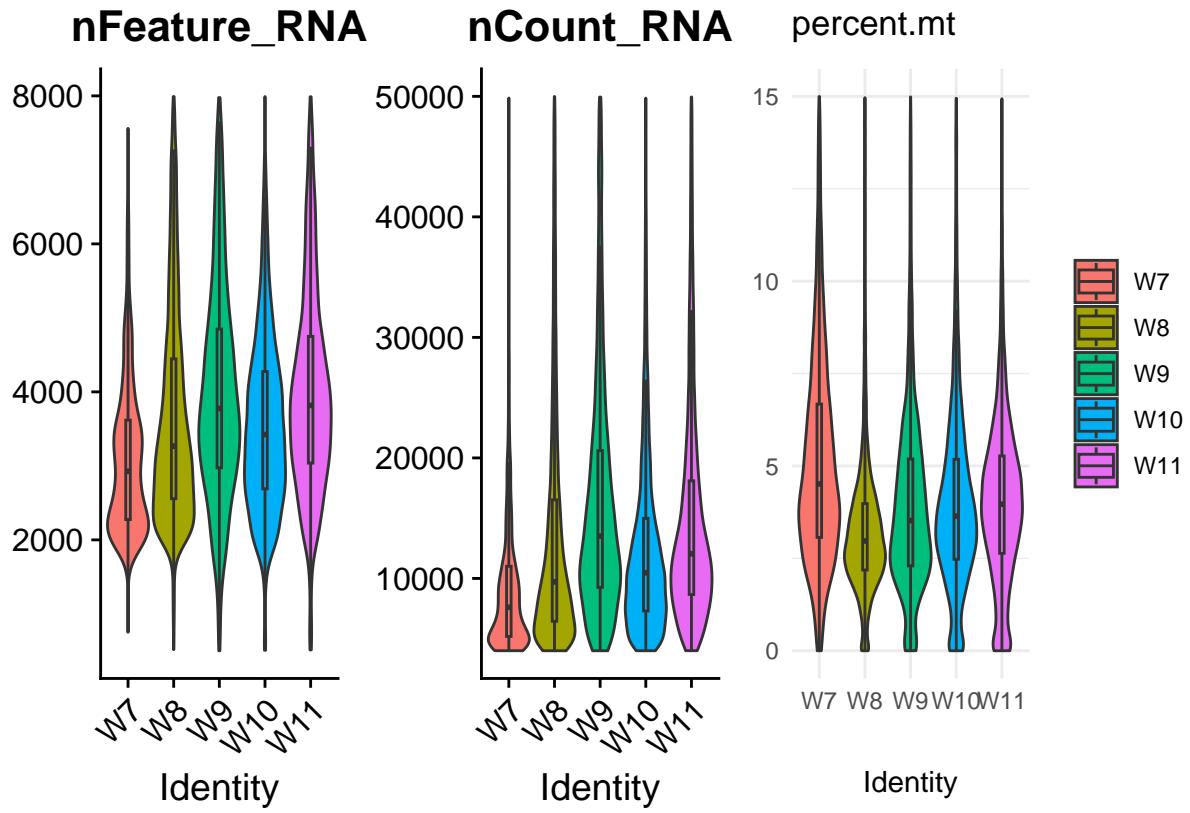
Hence, we don't do filtering again.

```
# Reorder the 'days' variable to follow the desired order
human7_11W$days <- factor(human7_11W$days, levels = c("W7", "W8", "W9", "W10", "W11"))

p <- VlnPlot(human7_11W, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"),
              ncol = 3, group.by = "days", pt.size = 0)

# Apply boxplot to each plot separately
p <- lapply(p, function(x) x + geom_boxplot(width = 0.1, outlier.shape = NA))

# Print the updated plots
patchwork::wrap_plots(p, ncol = 3) + theme_minimal()
```



```
# Add a column "time" to mark object source for integration
human7_11W$time <- "7_11"
```

## human8\_20W

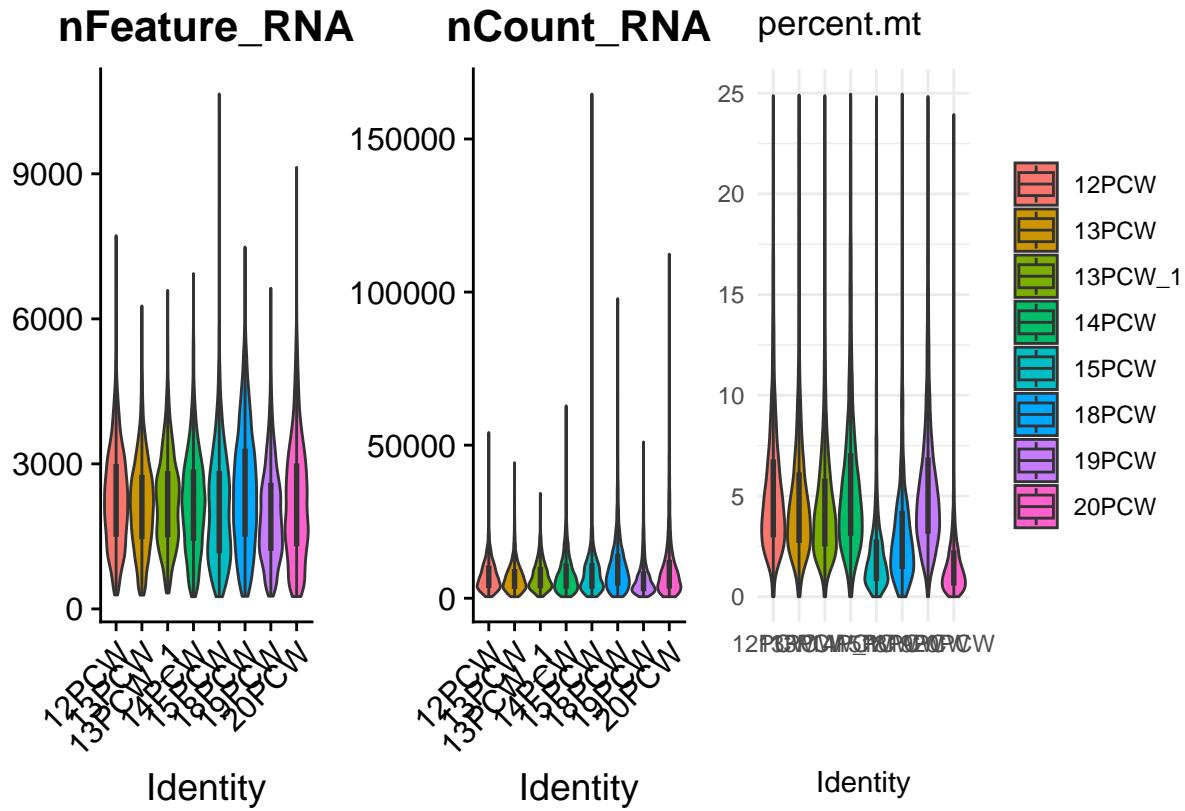
The Seurat object is filtered by nCount\_RNA > 500 & nFeature\_RNA > 250 & log10GenesPerUMI > 0.75 & percent.mt < 25

```
human12_20W<- subset(human12_20W, subset = nCount_RNA > 500 & nFeature_RNA > 250 & log10GenesPerUMI > 0.75 & percent.mt < 25)

p <- VlnPlot(human12_20W, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"),
              ncol = 3, group.by = "Gestation_Age", pt.size = 0)

# Apply boxplot to each plot separately
p <- lapply(p, function(x) x + geom_boxplot(width = 0.1, outlier.shape = NA))

# Print the updated plots
patchwork::wrap_plots(p, ncol = 3) + theme_minimal()
```



```
# Add a column "time" to mark object source for integration
human12_20W$time <- "12_20"
```

The dimension of each Seurat object

```
print("human4_6W")
## [1] "human4_6W"

print(paste("Number of genes:", dim(human4_6W)[1], "; Number of cells:", dim(human4_6W)[2]))
## [1] "Number of genes: 23093 ; Number of cells: 35079"

print("human7_11W")
## [1] "human7_11W"

print(paste("Number of genes:", dim(human7_11W)[1], "; Number of cells:", dim(human7_11W)[2]))
## [1] "Number of genes: 23663 ; Number of cells: 41727"
```

```

print("human12_20W")

## [1] "human12_20W"

print(paste("Number of genes:", dim(human12_20W)[1], "; Number of cells:", dim(human12_20W)[2]))

## [1] "Number of genes: 3000 ; Number of cells: 24342"

```

## Unify the format of 3 Seurat objects

```

library(dplyr)

# Modify the "Gestation_Age" column in metadata
human12_20W@meta.data$Gestation_Age <- human12_20W@meta.data$Gestation_Age %>%
  as.character() %>% # Ensure it's a character vector
  gsub("PCW", "", .) %>% # Remove "PCW"
  paste0("W", .) # Add "W" at the beginning

# Convert to a factor with a specific order
human12_20W@meta.data$Gestation_Age <- factor(human12_20W@meta.data$Gestation_Age,
                                                levels = c("W12", "W13", "W13_1", "W14",
                                                          "W15", "W18", "W19", "W20"))

# Set the new identity
Idents(human12_20W) <- human12_20W@meta.data$Gestation_Age

# Rename "Gestation_Age" with "days"
colnames(human12_20W@meta.data)[colnames(human12_20W@meta.data) == "Gestation_Age"] <- "days"

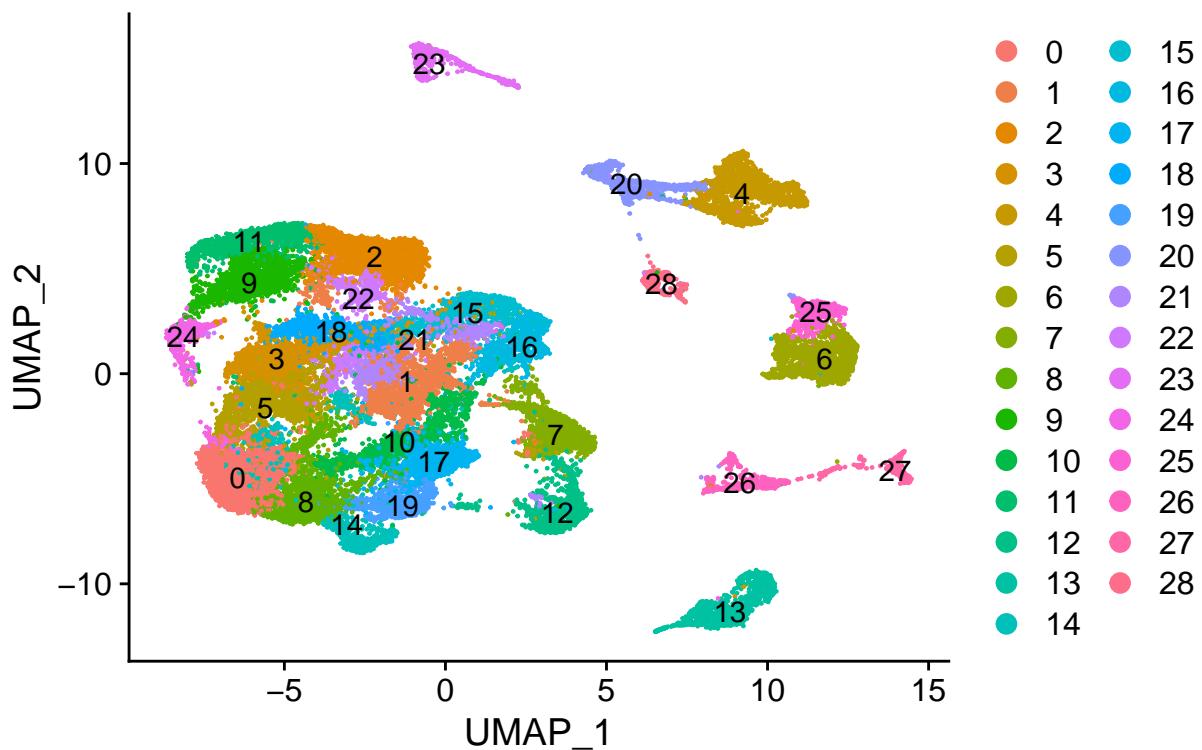
four_six <- RunPCA(human4_6W, verbose = FALSE)
four_six <- RunUMAP(human4_6W, dims = 1:30, verbose = FALSE)

## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'
## This message will be shown once per session

four_six <- FindNeighbors(human4_6W, dims = 1:30, verbose = FALSE)
four_six <- FindClusters(human4_6W, verbose = FALSE)
DimPlot(four_six, label = TRUE) + ggtitle("UMAP for human4_6W")

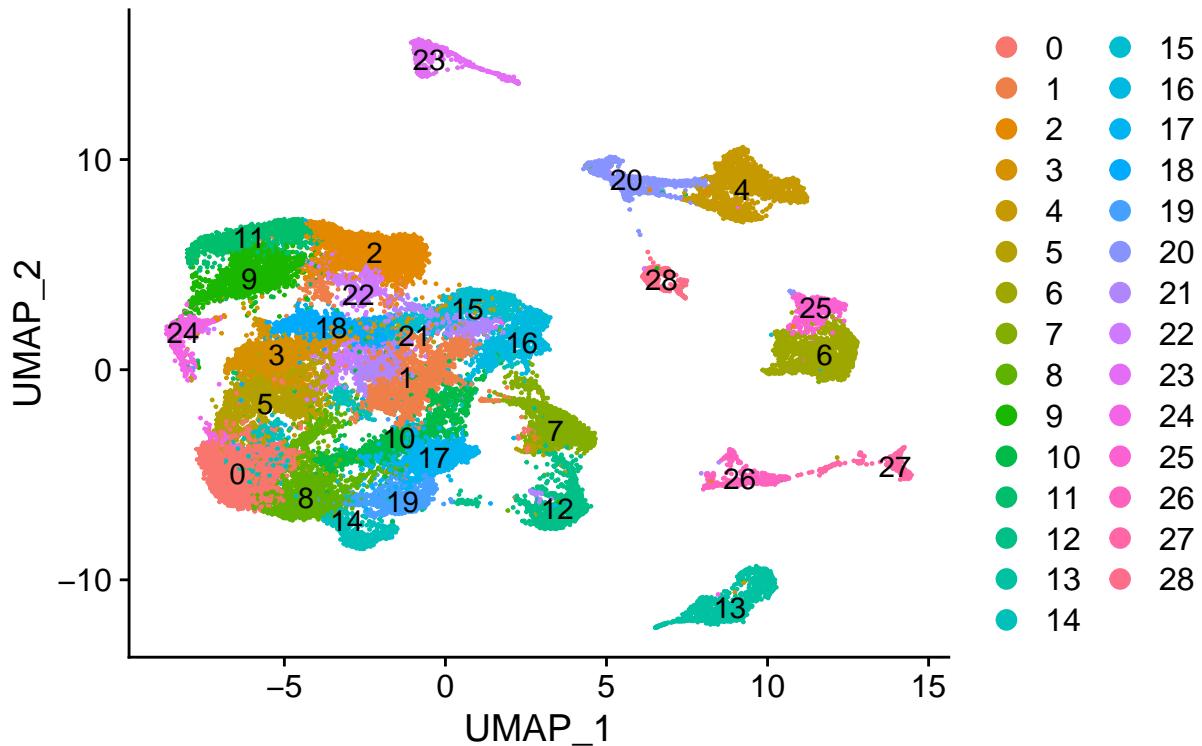
```

## UMAP for human4\_6W



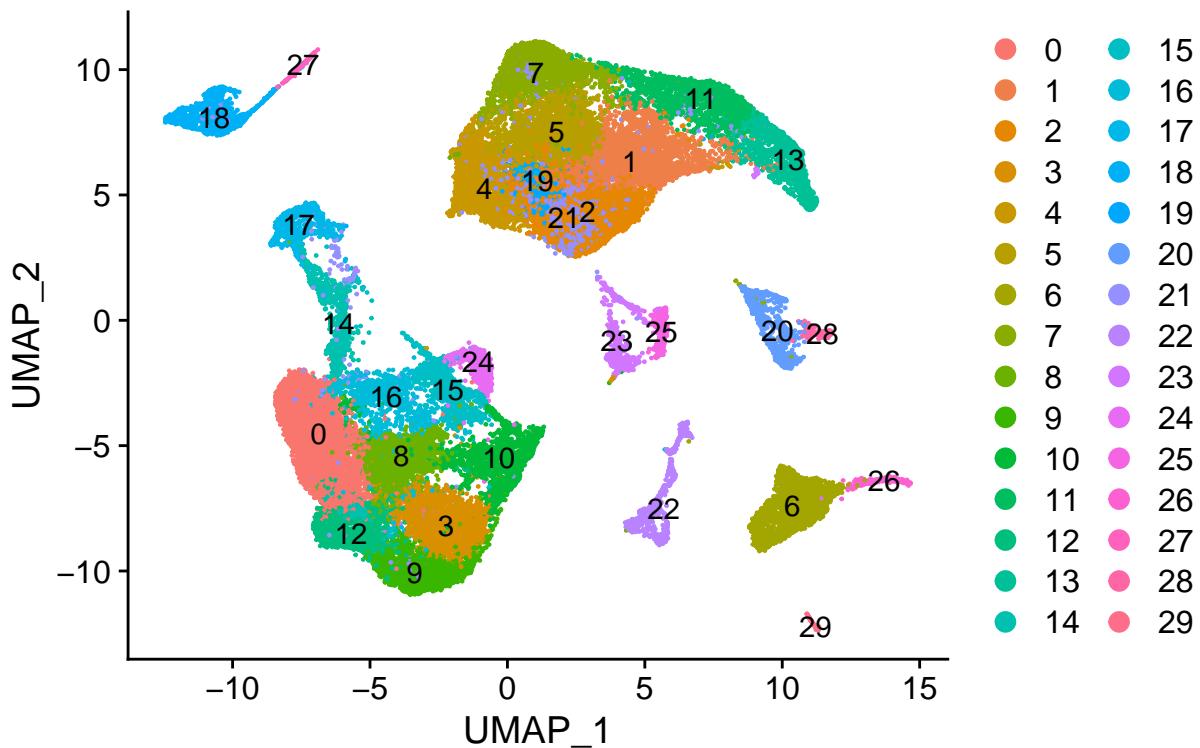
```
ggsave("../outputs/UMAP4_6W.png",
       plot = DimPlot(four_six, label = TRUE) + ggtitle("UMAP for human4_6W"),
       width = 8, height = 6, dpi = 300)
DimPlot(four_six, label = TRUE) + ggtitle("UMAP for human4_6W")
```

## UMAP for human4\_6W



```
seven_eleven <- RunPCA(human7_11W, verbose = FALSE)
seven_eleven <- RunUMAP(human7_11W, dims = 1:30, verbose = FALSE)
seven_eleven <- FindNeighbors(human7_11W, dims = 1:30, verbose = FALSE)
seven_eleven <- FindClusters(human7_11W, verbose = FALSE)
ggsave("../outputs/UMAP7_11W.png",
       plot = DimPlot(seven_eleven, label = TRUE) + ggtitle("UMAP for human7_11W"),
       width = 8, height = 6, dpi = 300)
DimPlot(seven_eleven, label = TRUE) + ggtitle("UMAP for human7_11W")
```

## UMAP for human7\_11W

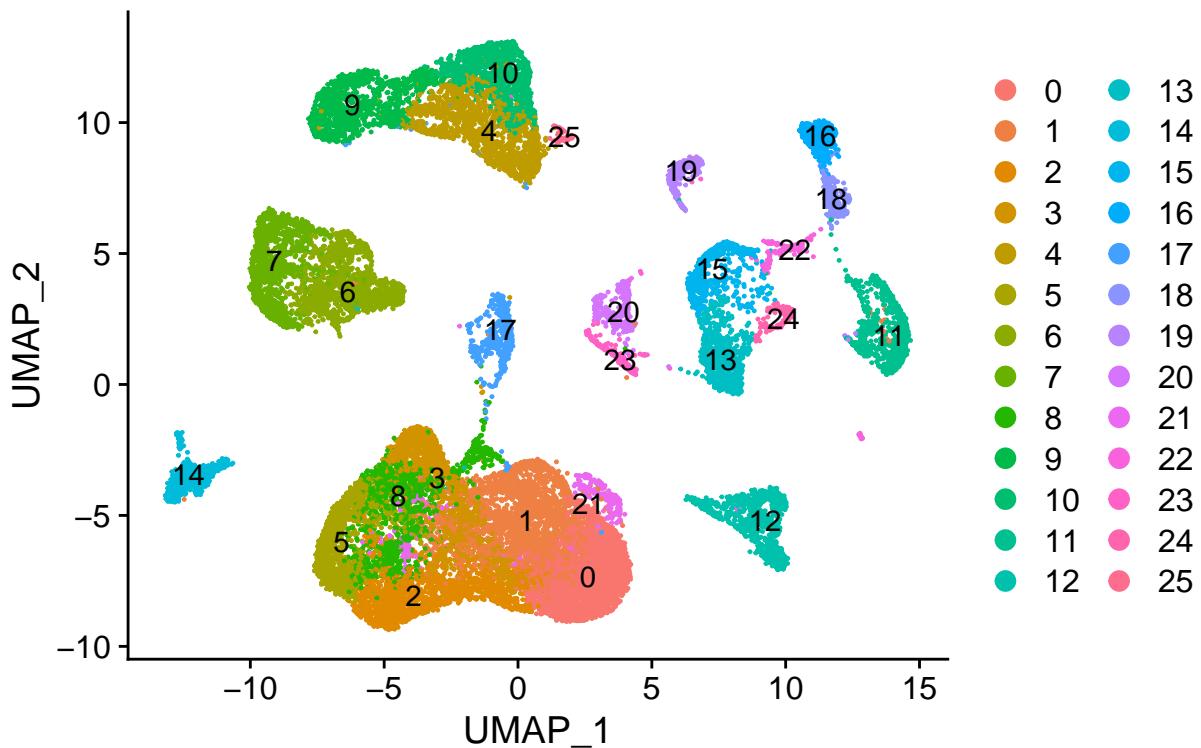


```
twelve_twenty <- RunPCA(human12_20W, verbose = FALSE)
twelve_twenty <- RunUMAP(human12_20W, dims = 1:30, verbose = FALSE)
twelve_twenty <- FindNeighbors(human12_20W, dims = 1:30, verbose = FALSE)
twelve_twenty <- FindClusters(human12_20W, verbose = FALSE)
```

```
## Warning: Adding a command log without an assay associated with it
```

```
ggsave("../outputs/UMAP12_20W.png",
       plot = DimPlot(twelve_twenty, label = TRUE) + ggtitle("UMAP for human12_20W"),
       width = 8, height = 6, dpi = 300)
DimPlot(twelve_twenty, label = TRUE)+ ggtitle("UMAP for human12_20W")
```

## UMAP for human12\_20W



Import the merged seurat object for downstream visualisations

```
library(ggplot2)

# Merge the Seurat objects
merged_object <- merge(x = human4_6W, y = list(human7_11W, human12_20W))
#saveRDS(merged_object, file = "../outputs/merged_object.rds")
merged_object <- readRDS("../outputs/merged_object.rds")
# Convert to a factor with a specific order
merged_object@meta.data$days <- factor(merged_object@meta.data$days,
                                         levels = c("W4", "W5", "W6",
                                                   "W7", "W8", "W9", "W10", "W11",
                                                   "W12", "W13", "W13_1", "W14", "W15", "W18", "W19",
                                                   "W20", "W21", "W22", "W23", "W24"))

# Plot violin plots
p <- VlnPlot(merged_object, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"),
              ncol = 1, group.by = "days", pt.size = 0)

# Convert to list and apply boxplot separately
plots <- lapply(p, function(x) x + geom_boxplot(width = 0.1, outlier.shape = NA, show.legend = FALSE))

# Save the plots
for (i in seq_along(plots)) {
```

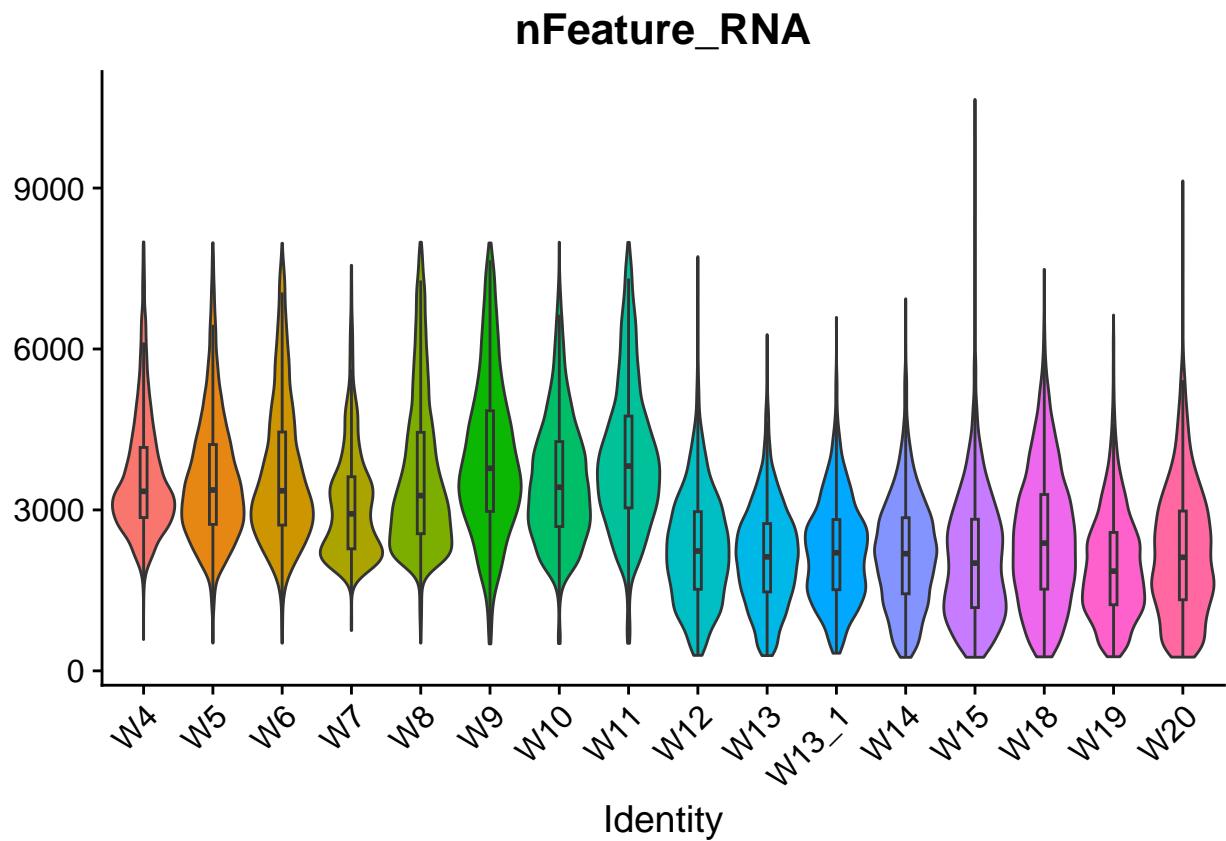
```

    ggsave(filename = paste0("../outputs/plot_", i, ".png"), plot = plots[[i]], width = 8, height = 6, dp
}

plots

## [[1]]

```

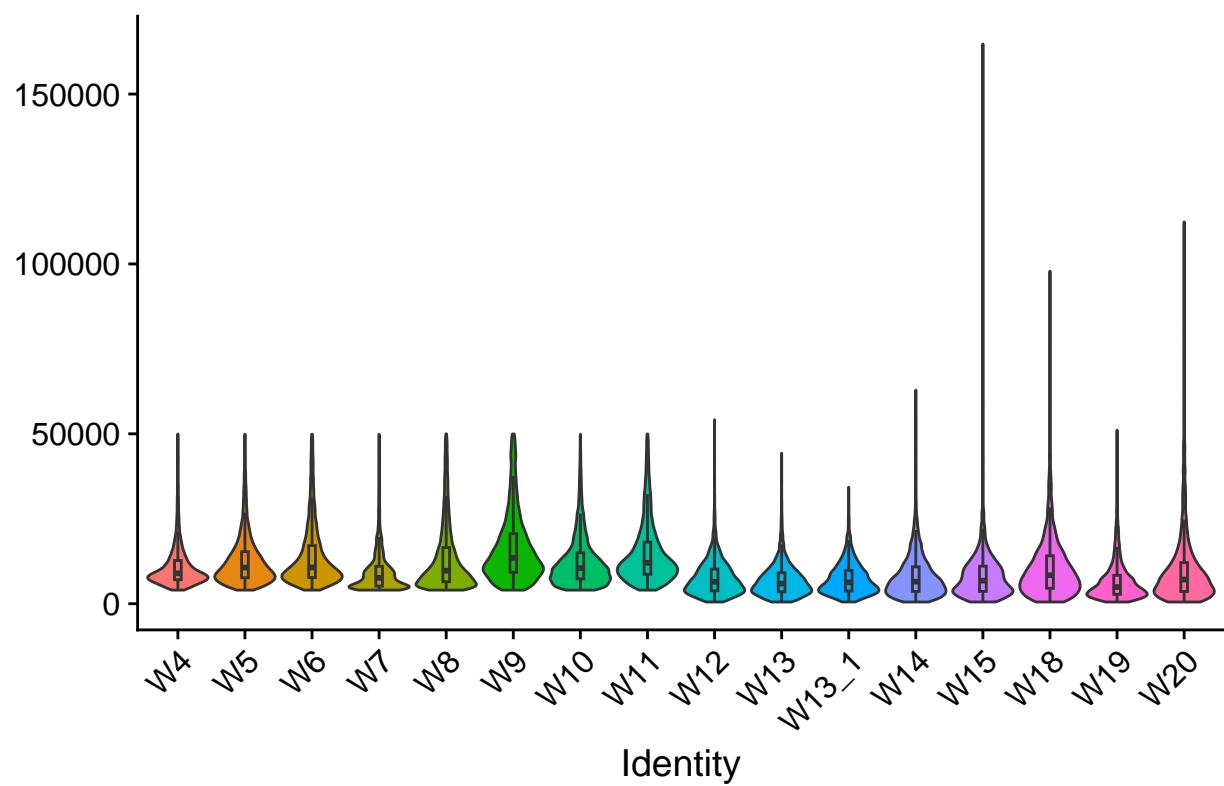


```

##
## [[2]]

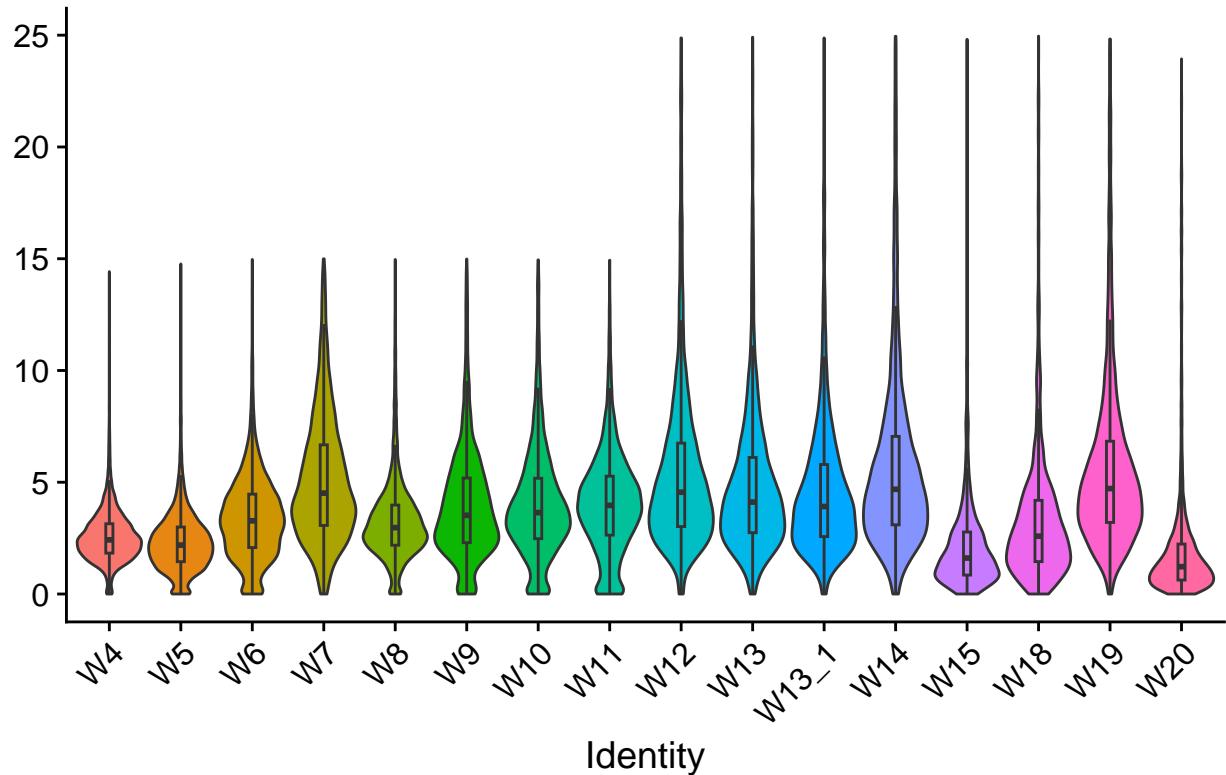
```

## nCount\_RNA



```
##  
## [[3]]
```

## percent.mt



```
## Bar plot
```

Visualize the number of cell counts in each period

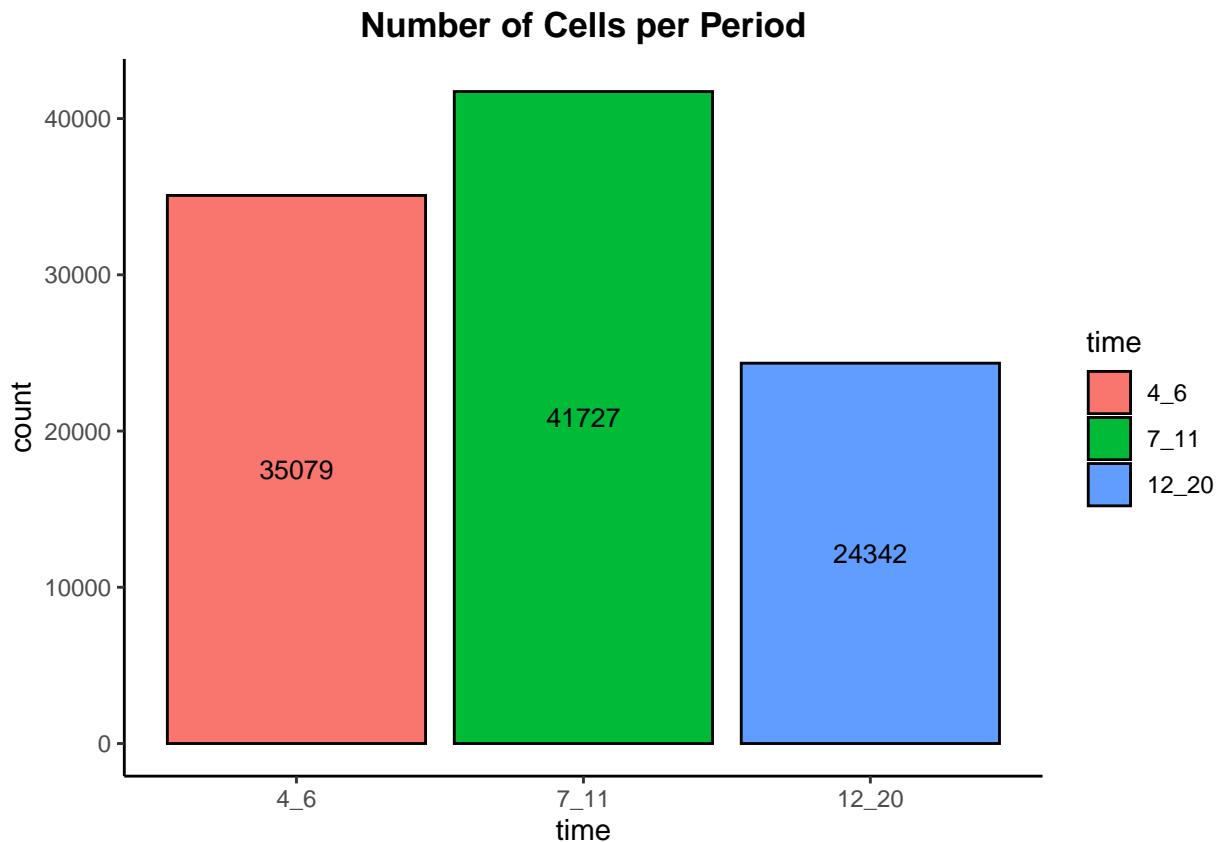
```
# Reorder the time in merged_object
merged_object$time <- factor(merged_object$time, levels = c("4_6", "7_11", "12_20"))

# Plot
plot <- merged_object@meta.data %>%
  ggplot(aes(x=time, fill=time)) +
  geom_bar(color="black") +
  stat_count(geom = "text", colour = "black", size = 3.5,
             aes(label = ..count..),
             position=position_stack(vjust=0.5))+
  theme_classic() +
  theme(plot.title = element_text(hjust=0.5, face="bold")) +
  ggtitle("Number of Cells per Period")

# Save the plot
ggsave("../outputs/nCellsPeriod.png", plot = plot,
       width = 8, height = 6, dpi = 300)
```

```
## Warning: The dot-dot notation ('..count..') was deprecated in ggplot2 3.4.0.
## i Please use 'after_stat(count)' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

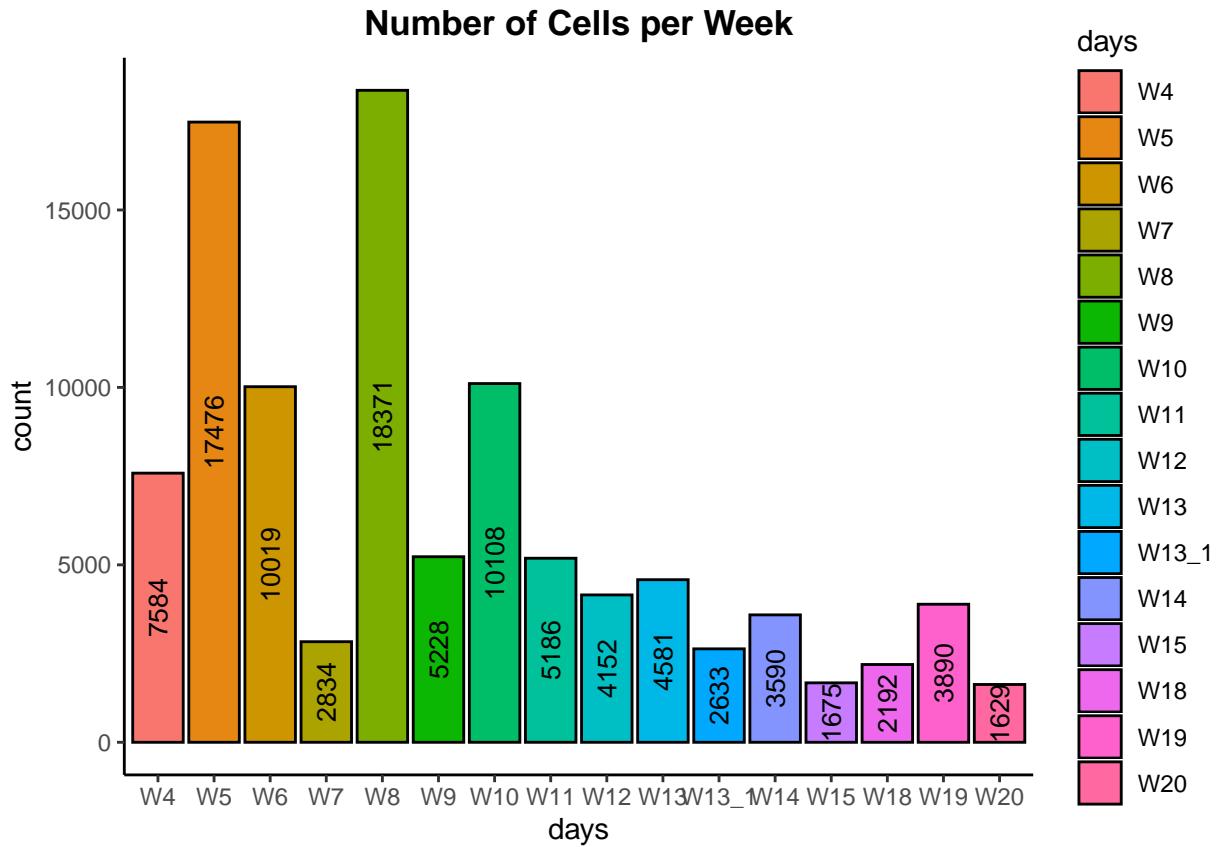
```
plot
```



```
# Plot
plot <- merged_object@meta.data %>%
  ggplot(aes(x=days, fill=days)) +
  geom_bar(color="black") +
  stat_count(geom = "text", colour = "black", size = 3.5,
             aes(label = ..count..),
             position=position_stack(vjust=0.5), angle = 90) +
  theme_classic() +
  theme(plot.title = element_text(hjust=0.5, face="bold")) +
  ggtitle("Number of Cells per Week")

# Save the plot
ggsave("../outputs/nCellsWeek.png", plot = plot,
       width = 8, height = 6, dpi = 300)

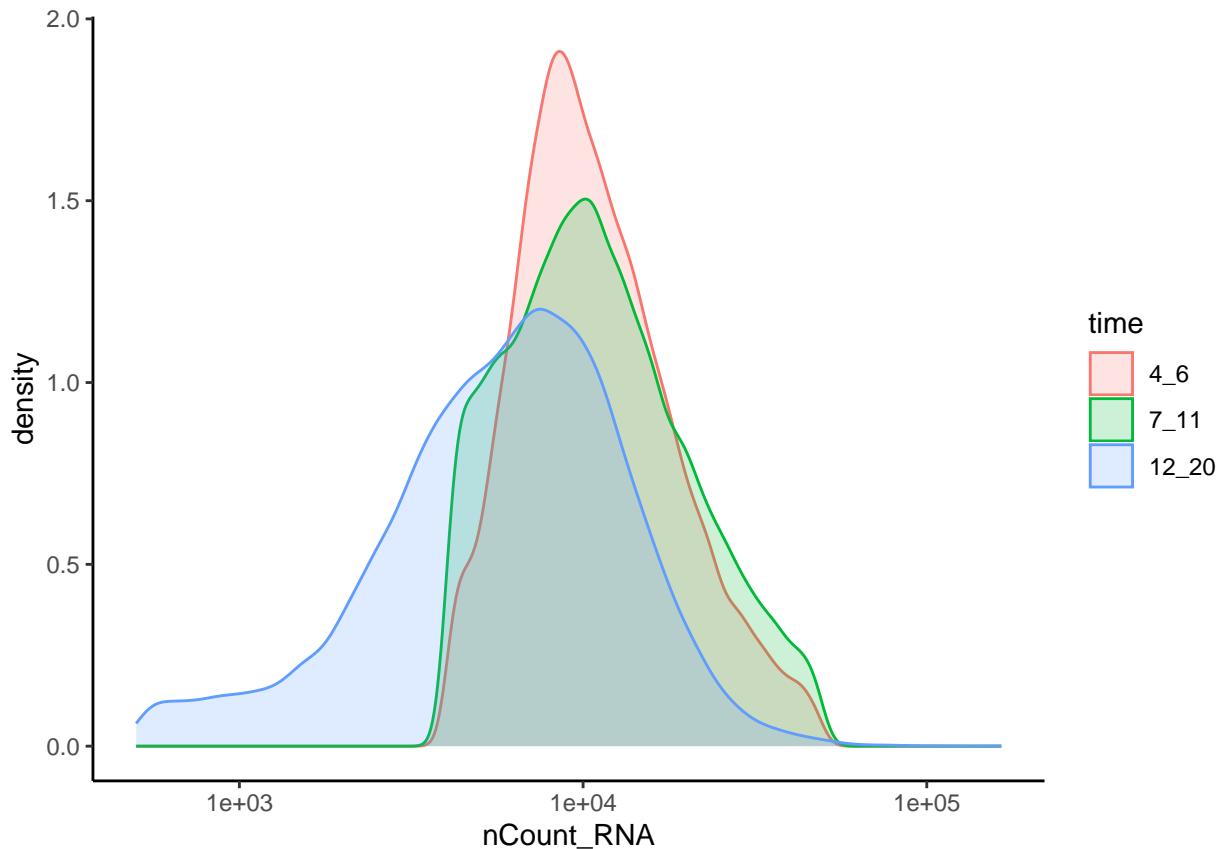
plot
```



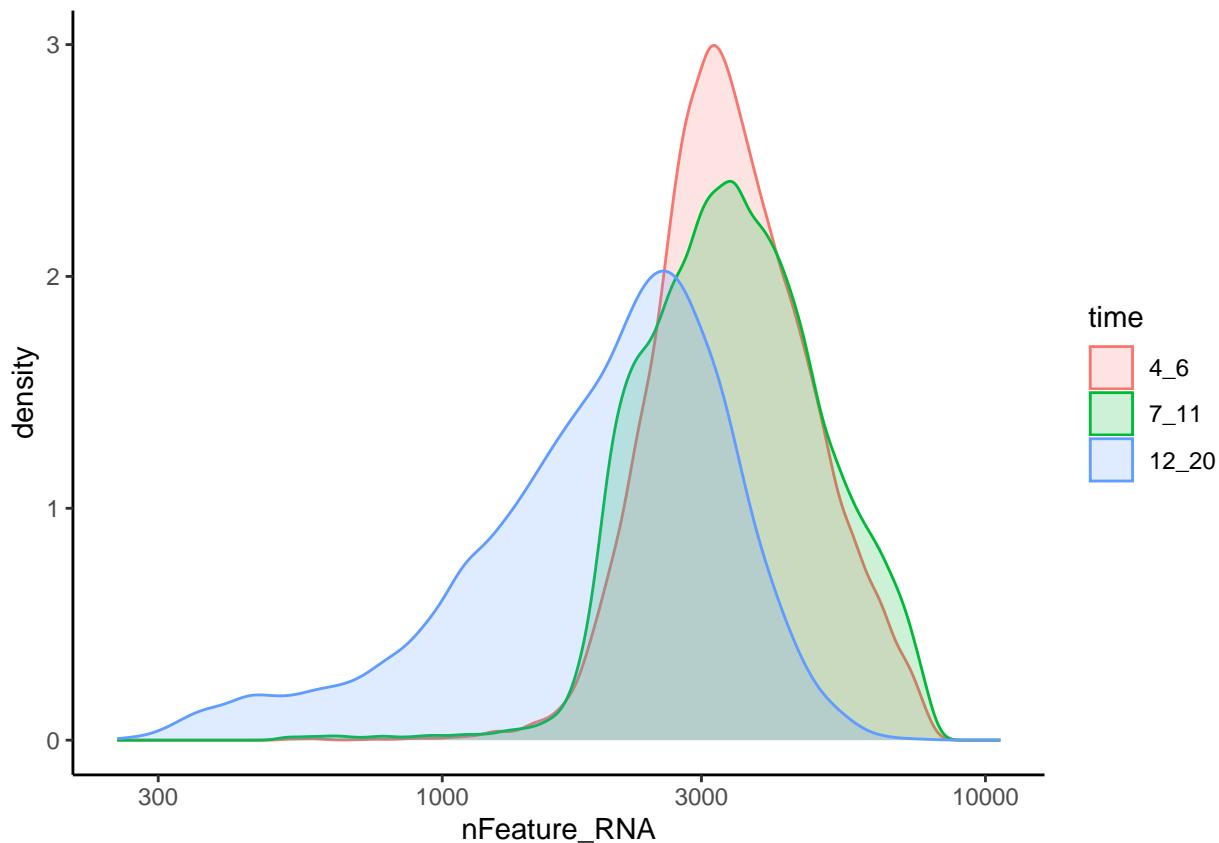
## Density Plot

Visualise the distribution of nCount, nFeature, and percent.mt between each seurat object

```
merged_object@meta.data %>%
  ggplot(aes(color=time, x=nCount_RNA, fill= time)) +
  geom_density(alpha = 0.2) +
  theme_classic() +
  scale_x_log10()
```



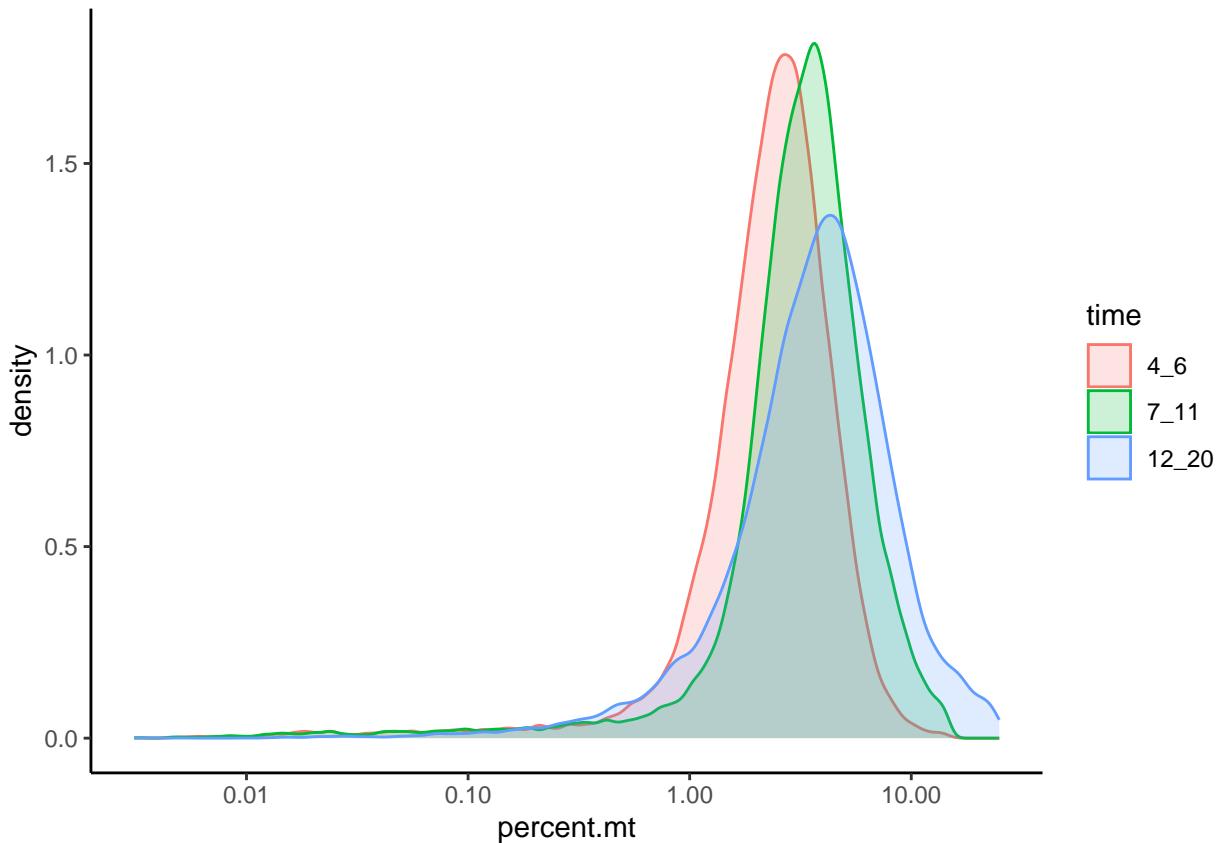
```
merged_object@meta.data %>%
  ggplot(aes(color=time, x=nFeature_RNA, fill= time)) +
  geom_density(alpha = 0.2) +
  theme_classic() +
  scale_x_log10()
```



```
merged_object@meta.data %>%
  ggplot(aes(color=time, x=percent.mt, fill= time)) +
  geom_density(alpha = 0.2) +
  theme_classic() +
  scale_x_log10()

## Warning in scale_x_log10(): log-10 transformation introduced infinite values.

## Warning: Removed 798 rows containing non-finite outside the scale range
## ('stat_density()'').
```



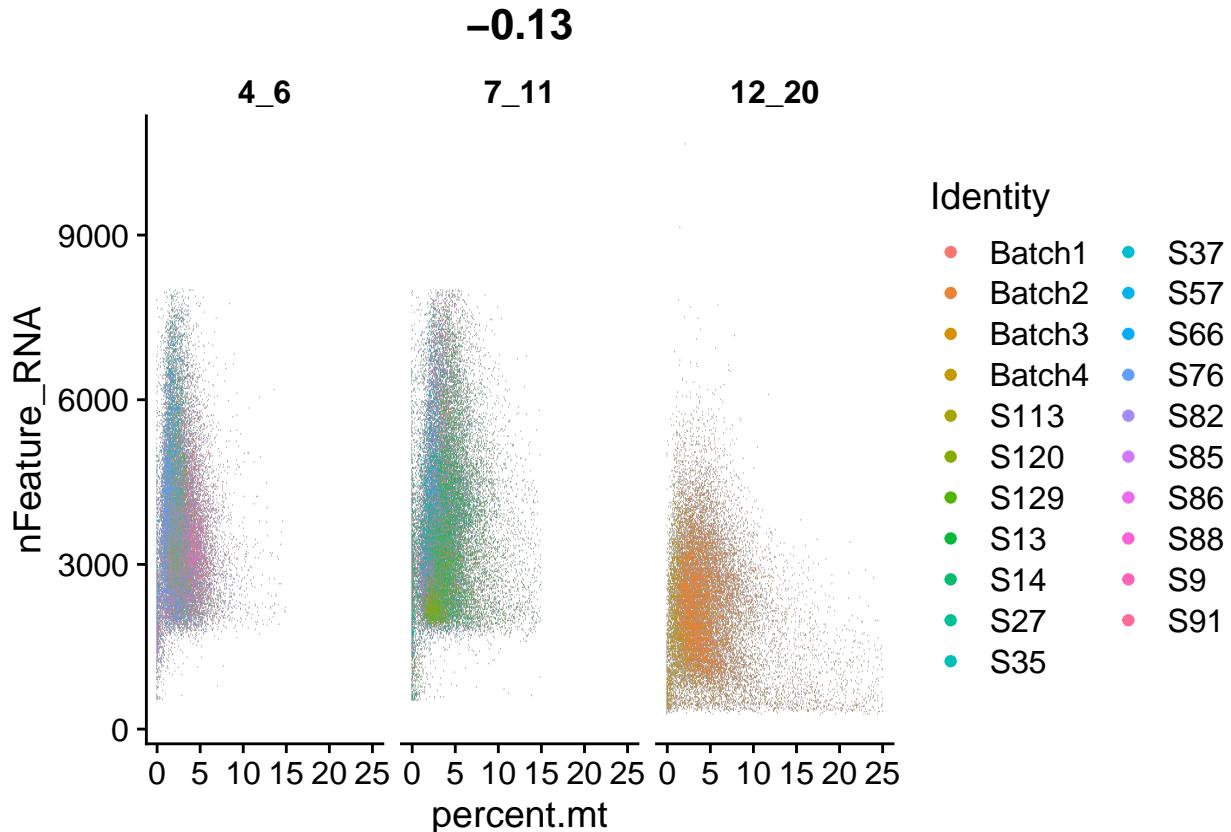
```
# geom_density: creates a density plot. alpha = transparency
# scale_x_log10: Transforms the x-axis to a logarithmic scale (base 10), so the large values are compressed
```

## Feature Scatter Plot

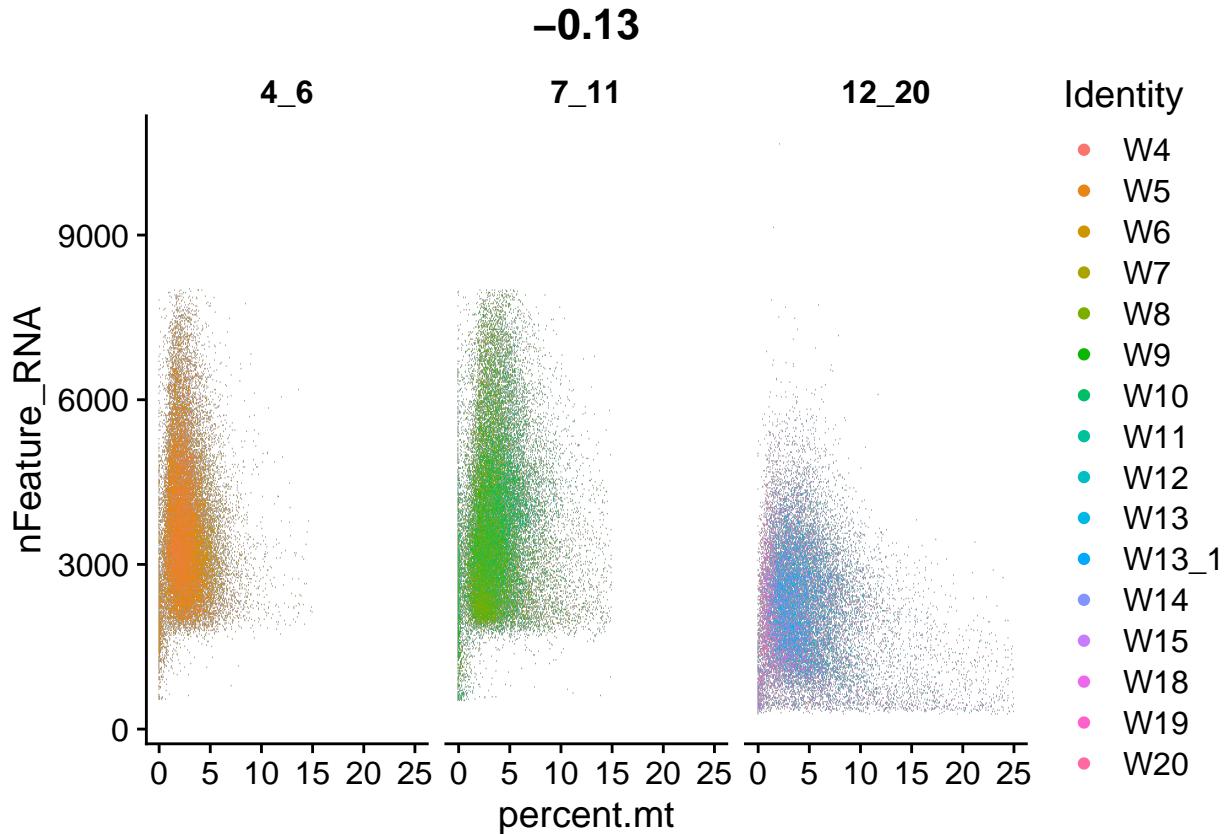
Scatter plot of percent.mt and nFeature

```
p <- FeatureScatter(merged_object, feature1 = "percent.mt", feature2 = "nFeature_RNA" , group.by="orig.id")
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'

ggsave("../outputs/FeaSca_mtF.png",
       plot = p,
       width = 12, height = 6, dpi = 300)
p
```



```
p <- FeatureScatter(merged_object, feature1 = "percent.mt", feature2 ="nFeature_RNA" , group.by="days",  
                        
## Rasterizing points since number of points exceeds 100,000.  
## To disable this behavior set 'raster=FALSE'  
  
ggsave("../outputs/FeaSca_mtF_days.png",  
       plot = p,  
       width = 12, height = 6, dpi = 300)  
p
```



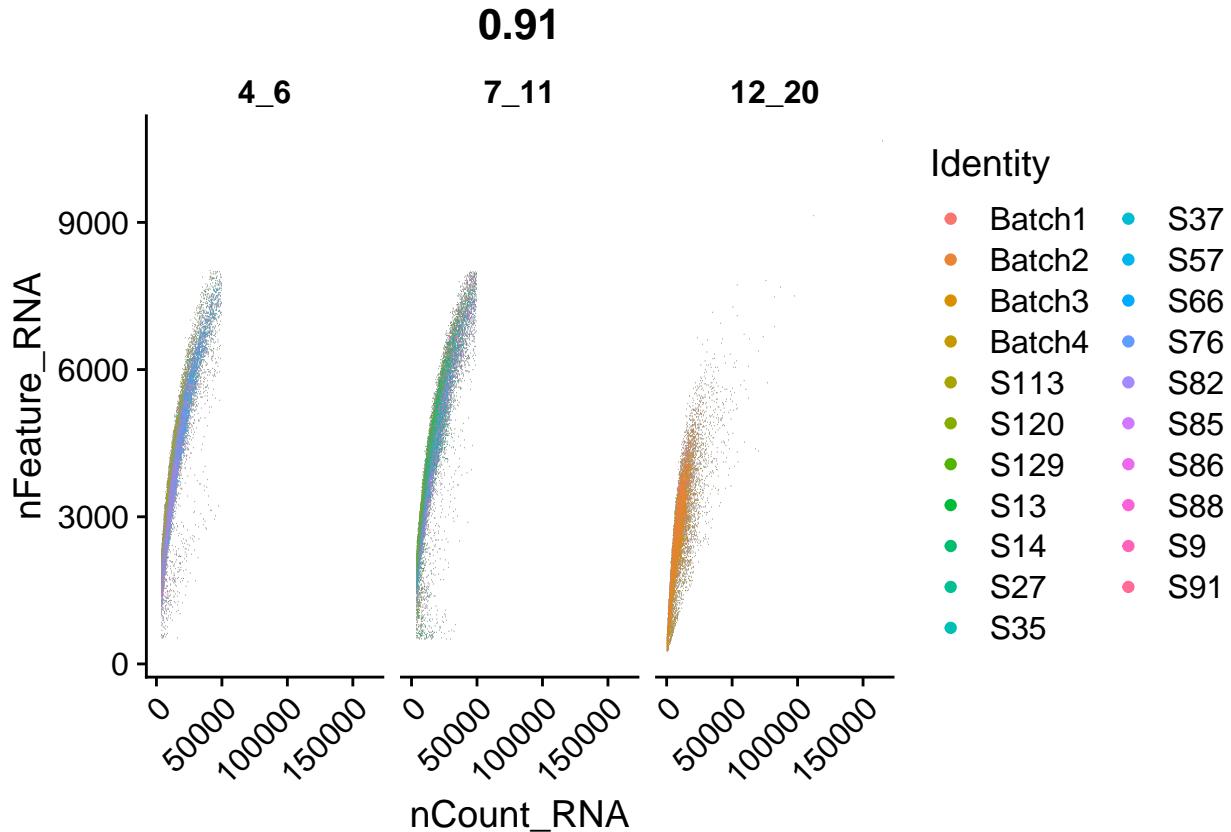
Scatter plot of nCount and nFeature

```
p <- FeatureScatter(merged_object, feature1 = "nCount_RNA", feature2 ="nFeature_RNA" , group.by="orig.id")

## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'

p <- p + theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggsave("../outputs/FeaSca_CF.png",
       plot = p,
       width = 12, height = 6, dpi = 300)
p
```



```

p <- FeatureScatter(merged_object, feature1 = "nCount_RNA", feature2 ="nFeature_RNA" , group.by="days", 
                     group.col="color", rasterize=TRUE)

## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'

p <- p + theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggsave("../outputs/FeaSca_CF_day.png",
       plot = p,
       width = 12, height = 6, dpi = 300)
p

```

