

# Untitled

Asad

5/4/2023

```
library(dplyr) library(tidyverse) library(ggplot2) library(ggpubr) library(RColorBrewer) library(wesanderson)
#Set directory setwd('D:/CancerData/TCGA LGG cBioPortal/LGG')
#Read RNA-seq data file and make data frame df<- read.table('data_mrna_seq_v2_rsem.txt', header =
T, sep = '\t') df<- data.frame(df)
#For example, we'll be working with TMED4 and TMED9 genes. So, let's filter those df2<- filter(df,
Hugo_Symbol=="TMED4" | Hugo_Symbol=="TMED9")
#Getting col data as row data df3<- data.frame(t(df2)) write.csv(df3, "Normalized Expression.csv")
#Let's prepare normalized and log2 groups normalized_expression<- read.csv("Normalized Expression.csv")
normalized_expressionPATIENT_ID <- gsub(".", "", as.character(normalized_expressionPATIENT_ID))
##Log2 Group log2_expression<- data.frame(log2(read.csv("Normalized Expression.csv", row.names =
1))) write.csv(log2_expression, "Log2 Expression.csv") log2_expression<- read.csv("Log2 Expression.csv")
log2_expressionPATIENT_ID <- gsub(".", "", as.character(log2_expressionPATIENT_ID))
#Getting clinical data and manipulating according to expression data clin_data<- data.frame(read.table('data_clinical_patie
header = T, sep = '\t')) clin_dataPATIENT_ID <- gsub("-", "", as.character(clin_dataPATIENT_ID))
#Merge two datasets Merged_normalized<- left_join(normalized_expression, clin_data, by="PATIENT_ID")
Merged_log2<- left_join(log2_expression, clin_data, by="PATIENT_ID")
#Save write.csv(Merged_normalized, "Merged_normalized.csv")
```

```
library(dplyr)
```

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

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

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

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.4.2      v purrr  1.0.1
## v tibble  3.2.1      v stringr 1.5.0
## v tidyr   1.3.0      v forcats 0.5.1
## v readr   2.1.2
```

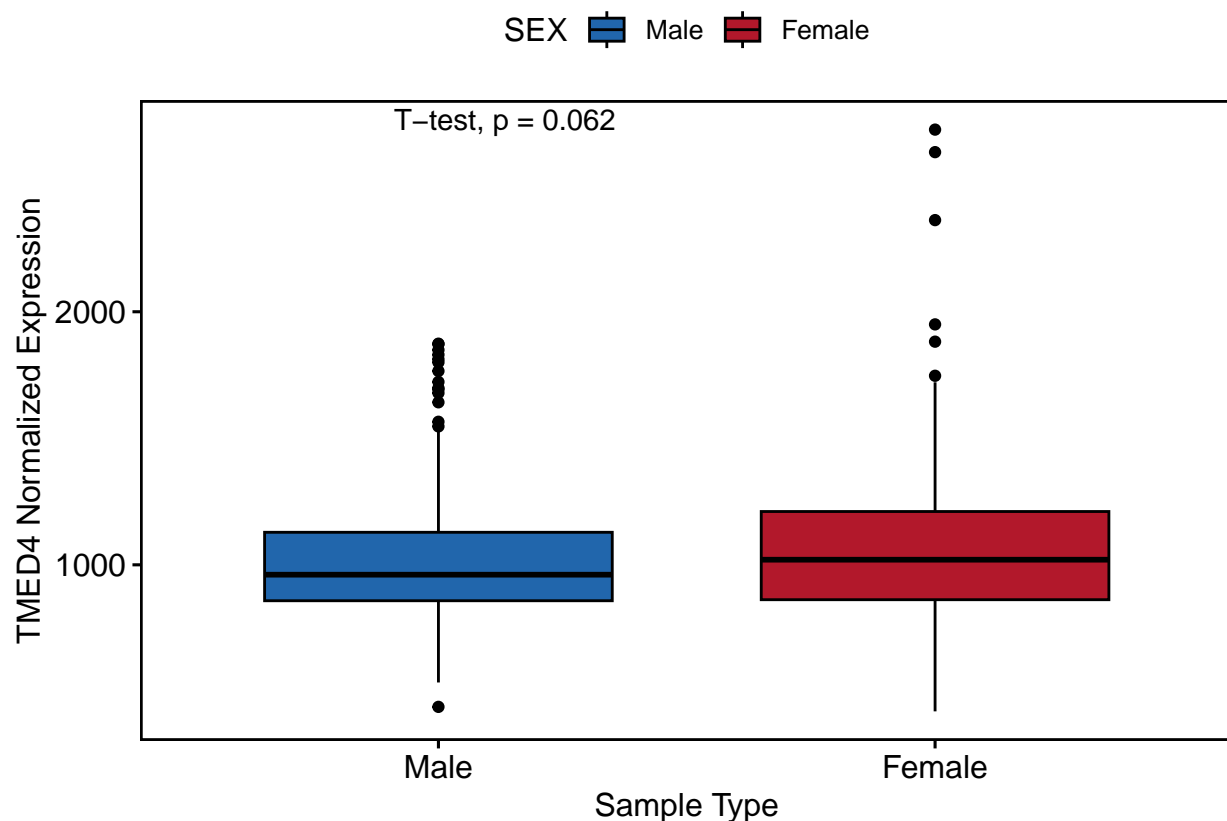
```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(ggpubr)
library(RColorBrewer)
library(wesanderson)

setwd('D:/CancerData/TCGA LGG cBioPortal/LGG')
Merged_normalized<-read.csv("Merged_normalized.csv")

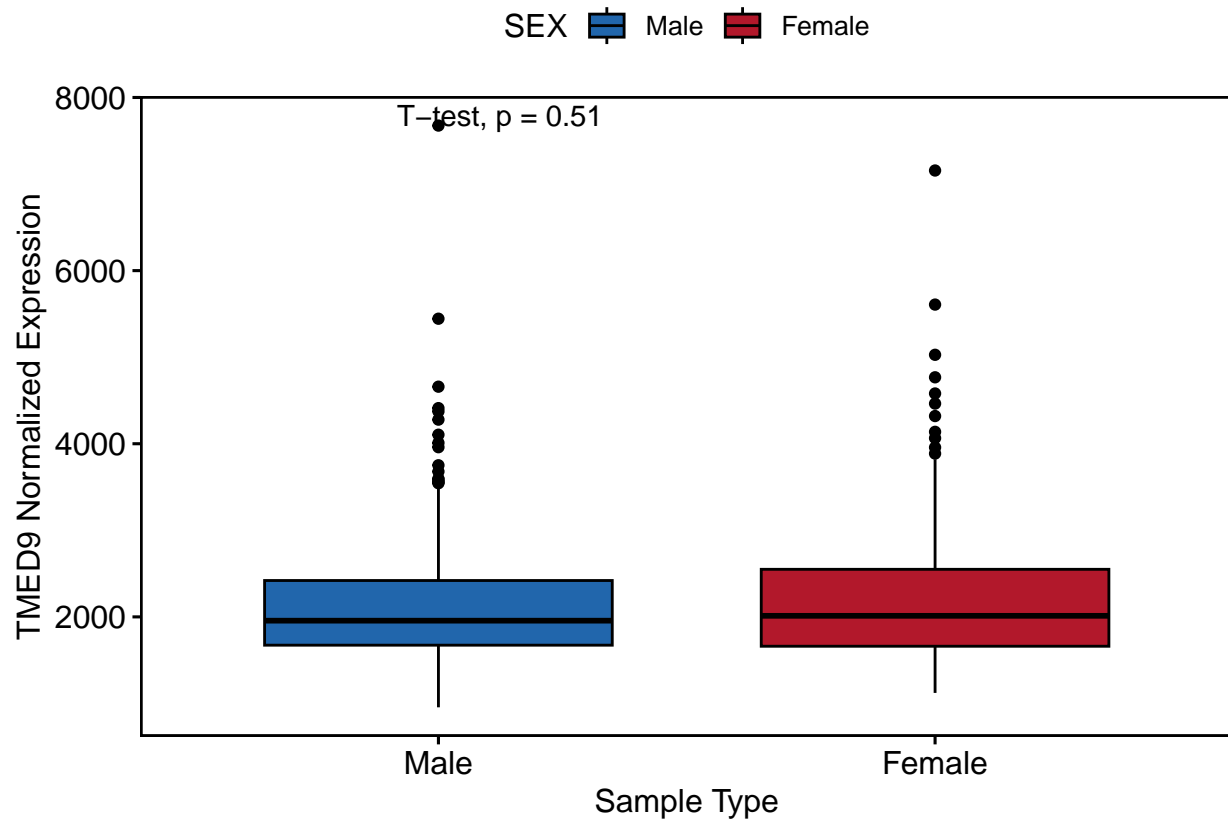
#Check expression pattern in accordance with patients' gender
gender<- na.omit(Merged_normalized[-402,c(2,3,8)]) #Remove NA and empty rows

TMED4_Gender <- ggboxplot(gender, x = "SEX", y = "TMED4",
  fill = "SEX", , palette = c("#2166AC", "#B2182B"),
  order = c("Male", "Female"), xlab = "Sample Type", ylab = "TMED4 Normalized Expression" ,
  stat_compare_means(method = 't.test', paired = F) + border()
TMED4_Gender
```



```
TMED9_Gender <- ggboxplot(gender, x = "SEX", y = "TMED9",
  fill = "SEX", , palette = c("#2166AC", "#B2182B"),
  order = c("Male", "Female"), xlab = "Sample Type", ylab = "TMED9 Normalized Expression",
  stat_compare_means(method = 't.test', paired = F) + border())
```

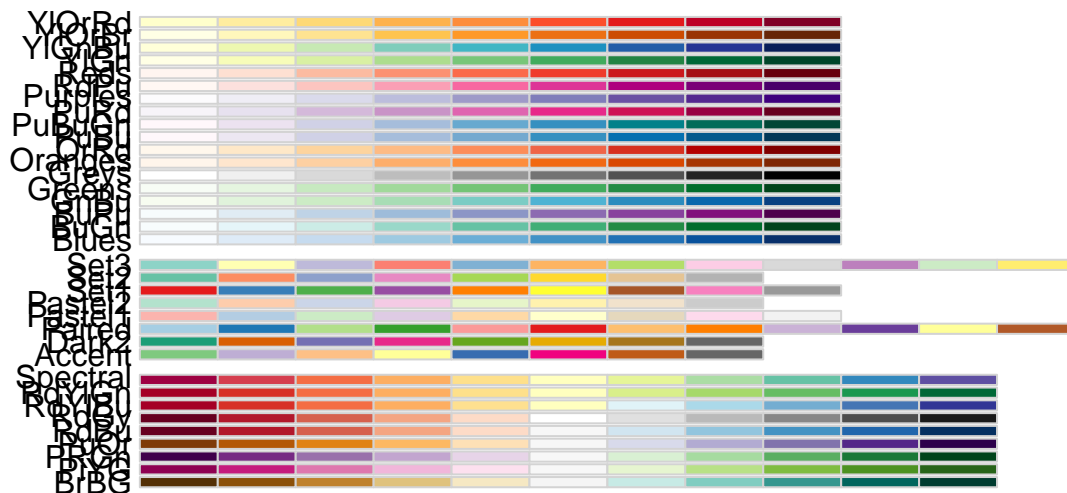
TMED9\_Gender



```
#Check expression pattern in accordance with cancer subtype
subtype<- na.omit(Merged_normalized[,2:4])
subtype[subtype==""]<- NA #Filling blank rows with NA values
subtype<- na.omit(subtype) #Removing NA values again

#Performing renaming of the variables
new.names<- c('LGG_IDHwt'='Wild Type', 'LGG_IDHmut-codel'='Mutant Codel',
  'LGG_IDHmut-non-codel' = 'Mutant non-Codel')
subtype$SUBTYPE<- new.names[subtype$SUBTYPE]

#Change color
display.brewer.all()
```



```
display.brewer.pal(n=10,name='Set1')
```

```
## Warning in display.brewer.pal(n = 10, name = "Set1"): n too large, allowed maximum for palette Set1 :
## Displaying the palette you asked for with that many colors
```

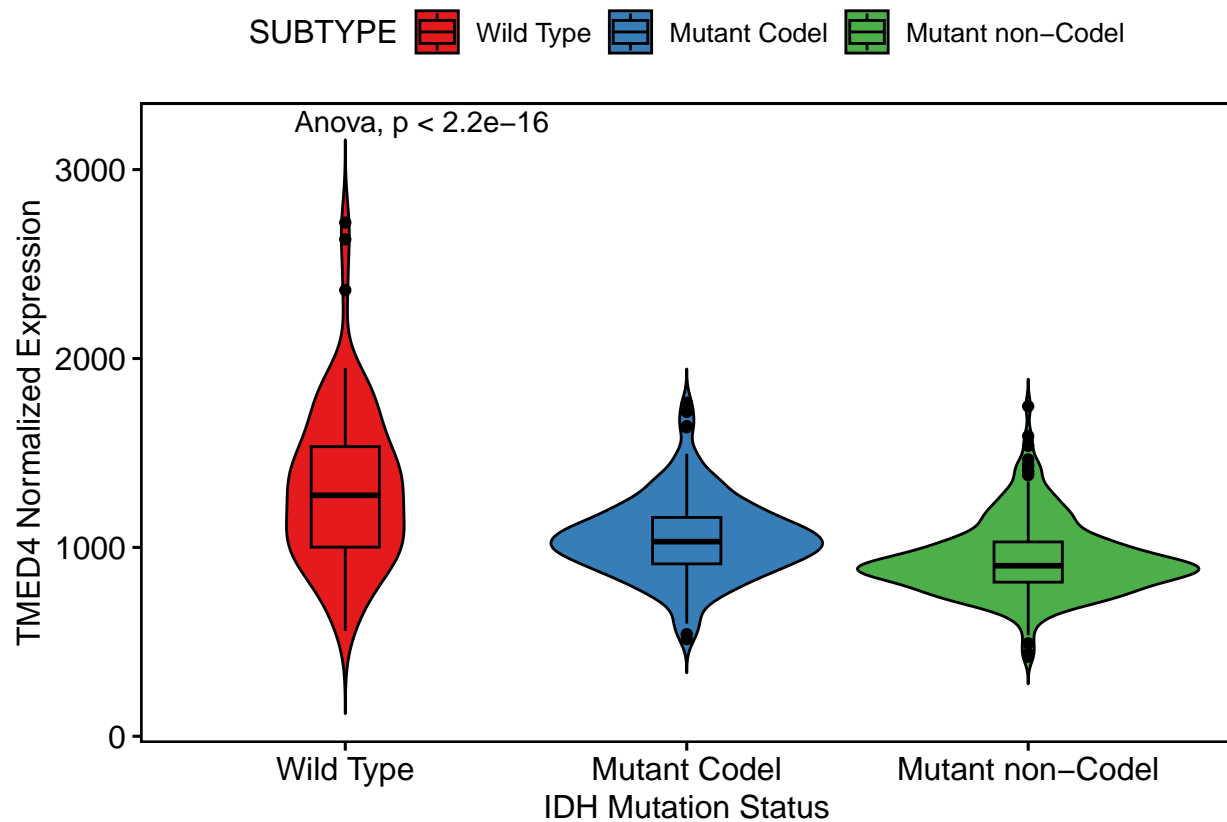


Set1 (qualitative)

```
TMED4_Subtype<- ggviolin(subtype, x = "SUBTYPE", y = "TMED4",  
                          fill = "SUBTYPE", palette = brewer.pal(n=10,name='Set1'),  
                          order = c('Wild Type','Mutant Codel', 'Mutant non-Codel'), add = 'boxplot',  
                          xlab = "IDH Mutation Status", ylab = "TMED4 Normalized Expression" ) +  
                          stat_compare_means(method = 'anova', label.y = 3200) + border()
```

```
## Warning in brewer.pal(n = 10, name = "Set1"): n too large, allowed maximum for palette Set1 is 9  
## Returning the palette you asked for with that many colors
```

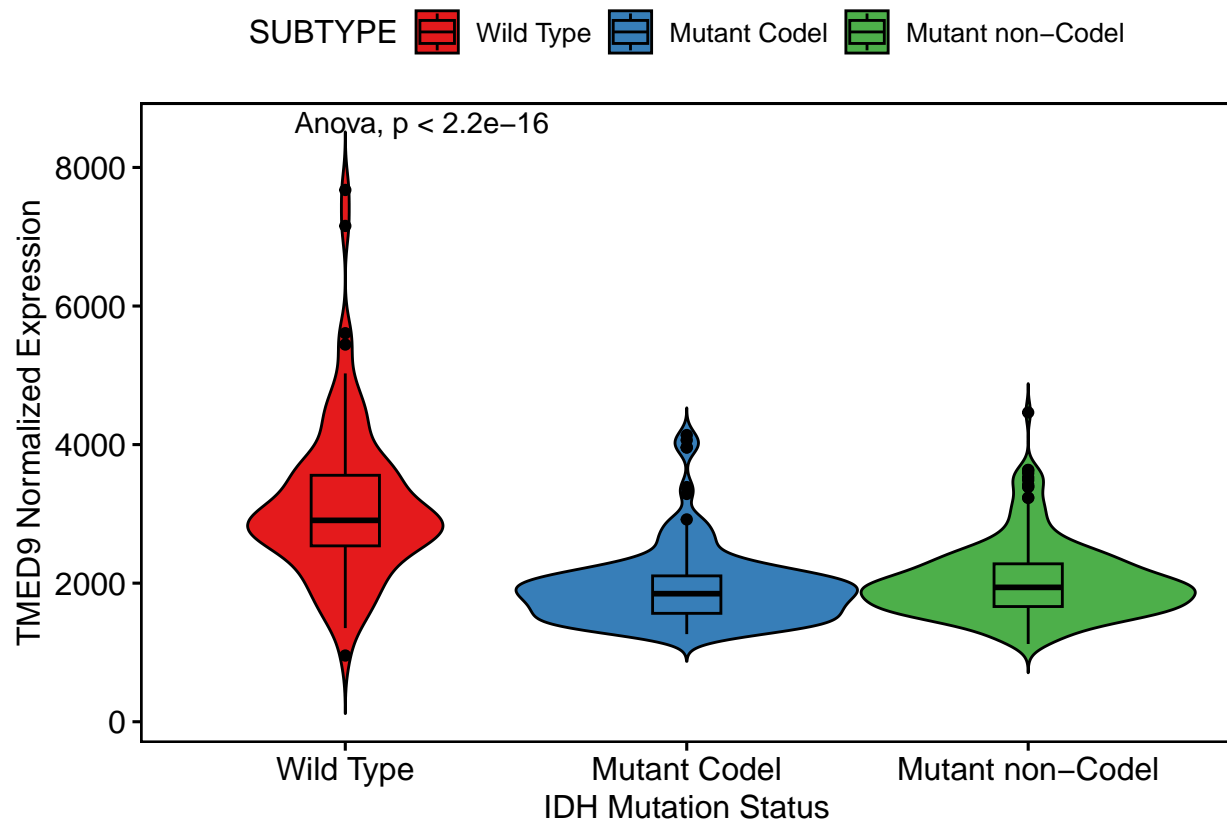
```
TMED4_Subtype
```



```
TMED9_Subtype<- ggviolin(subtype, x = "SUBTYPE", y = "TMED9",
                          fill = "SUBTYPE", palette = brewer.pal(n=10,name='Set1'),
                          order = c('Wild Type','Mutant Codel', 'Mutant non-Codel'), add = 'boxplot',
                          xlab = "IDH Mutation Status", ylab = "TMED9 Normalized Expression" ) +
  stat_compare_means(method = 'anova', label.y = 8500) + border()
```

```
## Warning in brewer.pal(n = 10, name = "Set1"): n too large, allowed maximum for palette Set1 is 9
## Returning the palette you asked for with that many colors
```

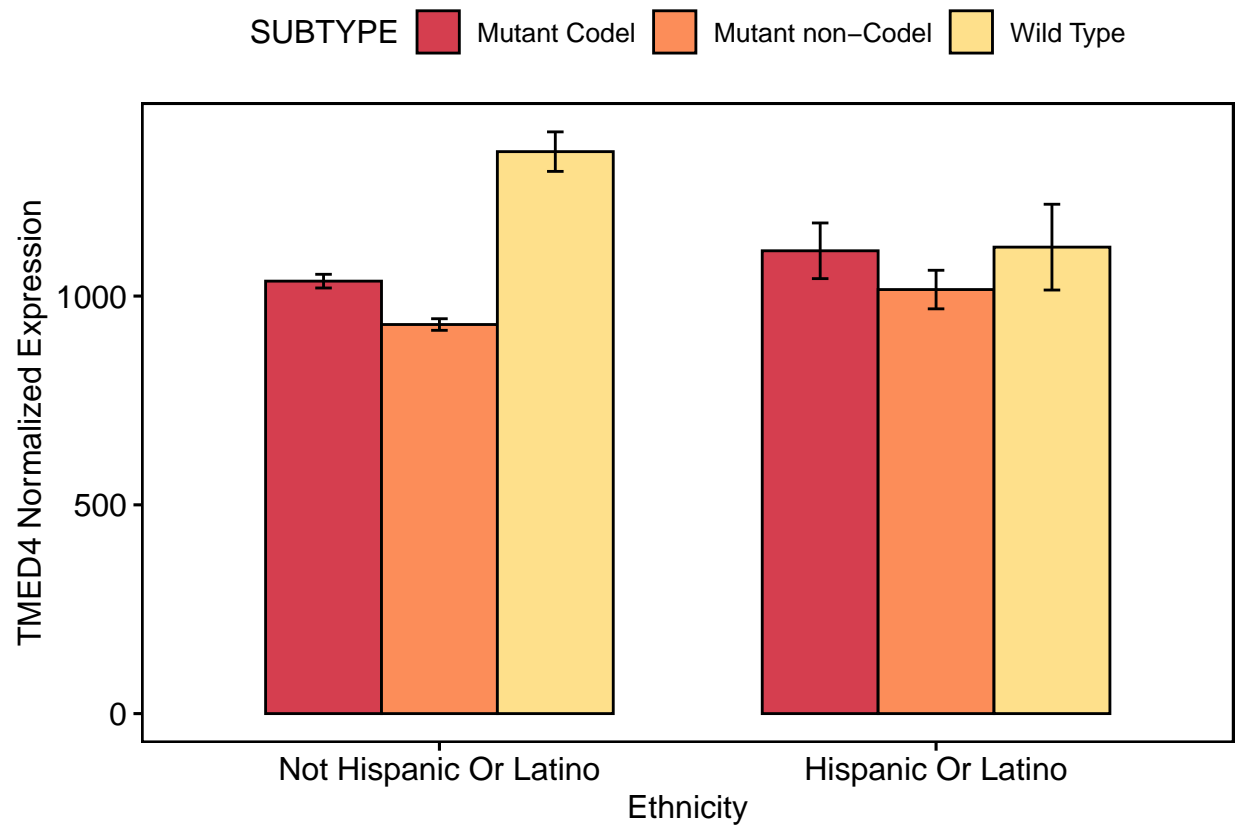
```
TMED9_Subtype
```



```
#Ethnicity
eth<- Merged_normalized[, c(2,3,4,14)]
eth[eth==""]<- NA #Filling blank rows with NA values
eth<- na.omit(eth) #Removing NA values again
eth$SUBTYPE<- new.names[eth$SUBTYPE]

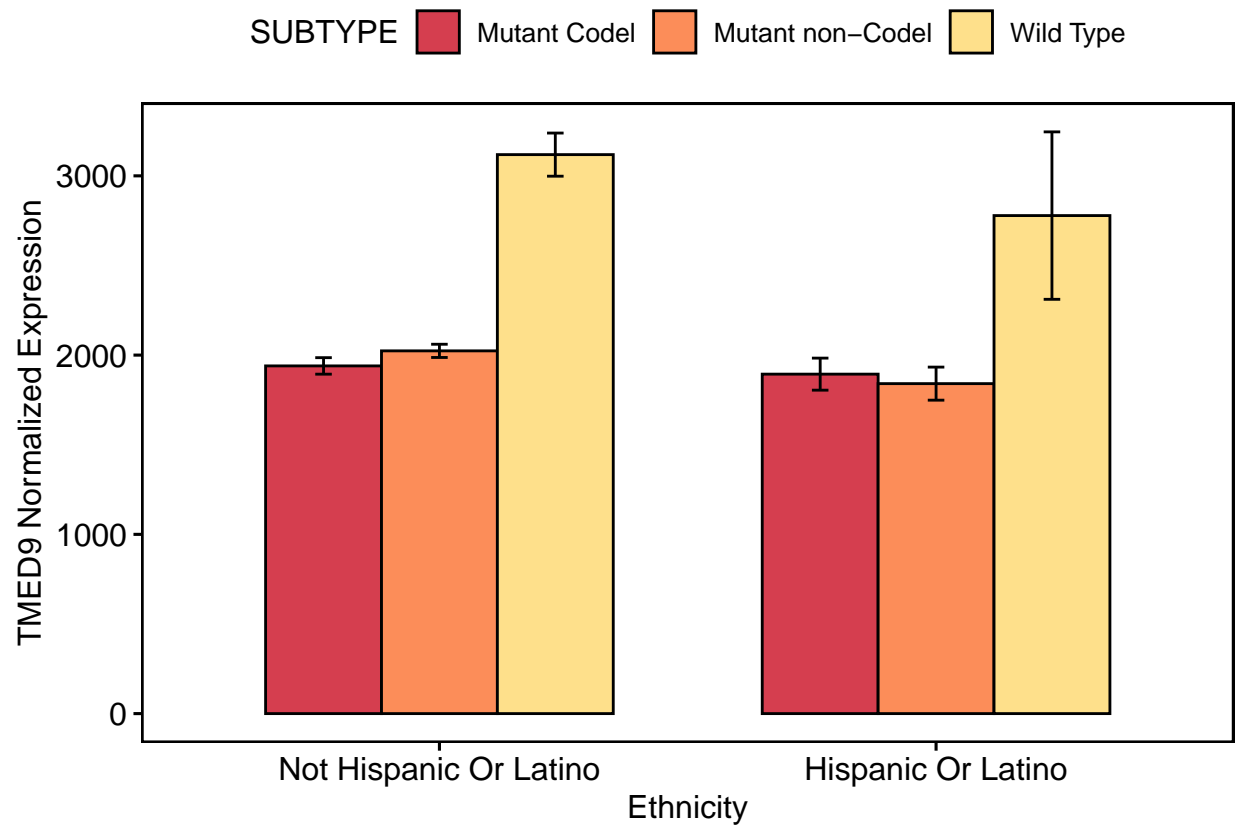
TMED4_eth<- ggbarplot(eth, x="ETHNICITY", y='TMED4', add = 'mean_se', fill = 'SUBTYPE',
  palette = brewer.pal(n = 7, name = "Spectral"), position = position_dodge(0.7),
  xlab = "Ethnicity", ylab = "TMED4 Normalized Expression")+border()

TMED4_eth
```



```
TMED9_eth<- ggbarplot(eth, x="ETHNICITY", y='TMED9', add = 'mean_se', fill = 'SUBTYPE',
  palette = brewer.pal(n = 7, name = "Spectral"), position = position_dodge(0.7),
  xlab = "Ethnicity", ylab = "TMED9 Normalized Expression")+border()
TMED9_eth
```





```
#Performing survival analysis  
Merged_log2<-read.csv("Merged_log2.csv")  
surv_data<- Merged_log2[, c(3,4,12,33,34)]  
write.csv(surv_data, 'surv_data.csv')
```