R Notebook

Attempting to model the S&P 500 annual returns (including dividends).

Lets read in the file:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

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

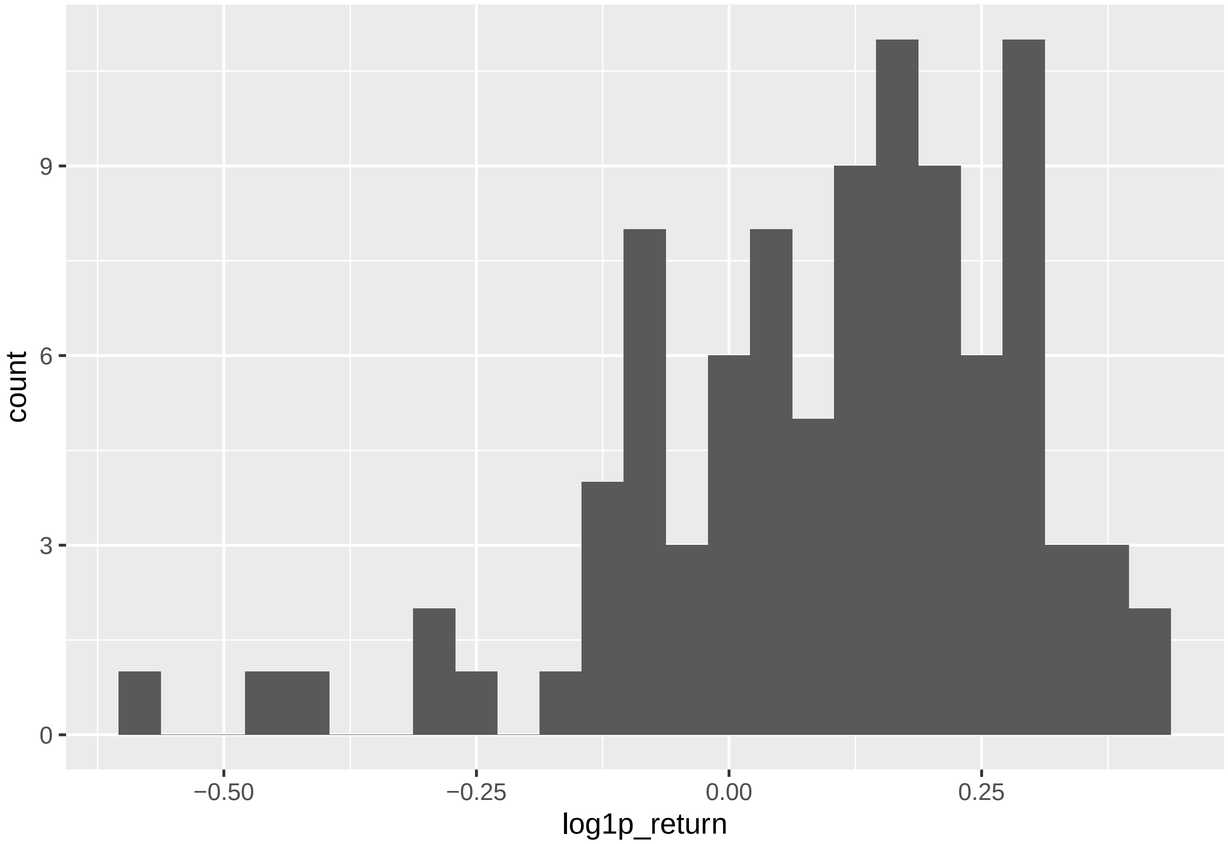
```
df_sp500 = readxl::read_excel('SP500_annual_returns.xlsx')
df_sp500 = df_sp500 %>%
  rename(return = '% Return', year = 'Year') %>%
  mutate(year = as.integer(year))
```

Now lets log transform it as returns are multiplicative, i.e., a return of -50% is not the same as +50% - -50% and +100% are equivalent, which we can model by log transforming.

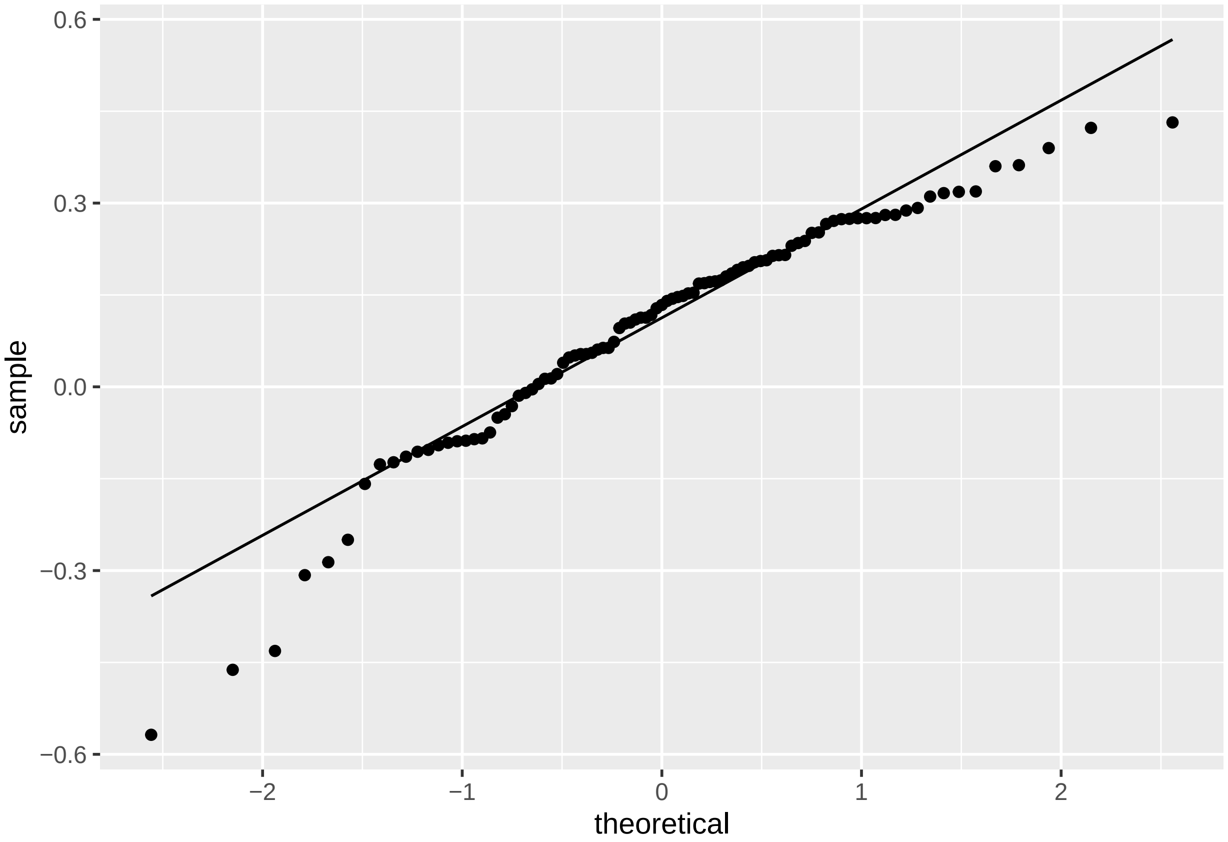
```
df_sp500 = df_sp500 %>% mutate(log1p_return = log1p(return/100.0))
```

And do some plots:

```
ggplot(data = df_sp500, aes(log1p_return)) + geom_histogram(bins = 25)
```



```
ggplot(data = df_sp500, aes(sample = log1p_return)) + stat_qq() + stat_qq_line()
```



Eyeballing it looks very highly skewed.

```
library(moments)
skewness(df_sp500 %>% pull(log1p_return))
```

```
## [1] -0.9940141
```

Let's calculate the moments, and use the sn package to create samples that return the log1p returns.

```
sp500_moments = list(
  mean = mean(df_sp500 %>% pull(log1p_return)),
  sd = sd(df_sp500 %>% pull(log1p_return)),
  skew = skewness(df_sp500 %>% pull(log1p_return))
)
```

```
library(sn)
```

```
## Loading required package: stats4
```

```
##
## Attaching package: 'sn'
```

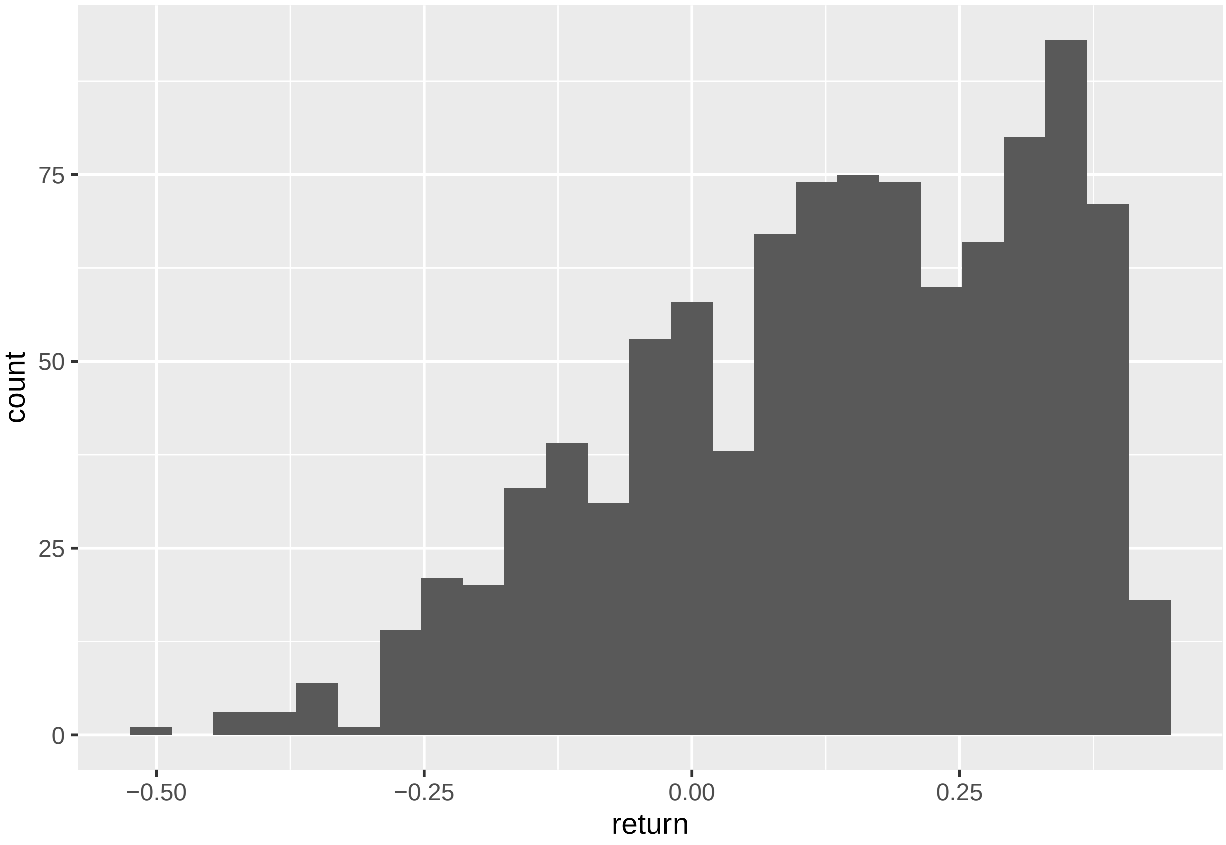
```
## The following object is masked from 'package:stats':
##
##     sd
```

```
sn_params = cp2dp(c(sp500_moments$mean, sp500_moments$sd, sp500_moments$skew), "SN")

df_returns = tibble(log1p_return = rsn(n=1000, dp = sn_params))

df_returns = df_returns %>% mutate(return = expm1(log1p_return))
```

```
df_returns
```

```
## # A tibble: 1,000 x 2
##    log1p_return  return
##           <dbl>   <dbl>
##  1       0.290   0.337
##  2      -0.224  -0.201
##  3       0.0135  0.0136
##  4      -0.309  -0.266
##  5       0.286   0.332
##  6      -0.0847 -0.0812
##  7       0.215   0.240
##  8       0.116   0.123
##  9       0.0577  0.0594
## 10       0.0484  0.0495
## # ... with 990 more rows
```

```
ggplot(data = df_returns, aes(return)) + geom_histogram(bins = 25)
```



And what about