Code

Teng Wei Yeo

```r
knitr::opts_chunk$set(fig.width=8, fig.height=5)
```

```r
##################
#### Overview ####
##################

## This code reproduces, in R, the averaged-ranks case in Fig. S2 of the
## following paper:

## K.A. McKinnon, I.R. Simpson, A.P. Williams,
## The pace of change of summertime temperature extremes,
## Proc. Natl. Acad. Sci. U.S.A.
## 121 (42) e2406143121,
## https://doi.org/10.1073/pnas.2406143121 (2024).

## The code produced and shared by the authors, "synthetic_data_tests.py",
## was used as a reference when reproducing the code.


## The last section of this code then tries a different test-case for the
## averaged-ranks case: instead of prescribing the variance to increase
## linearly over the 65 years, it instead enforces the variance
## to increase exponentially over the 65 years.
## The key steps are as follows:
# var_seq <- seq(0, 0.5, 0.05) ## Changes in the variance tried
# exp_seq <- c(0, exp(1:(years-1)) / exp((years-1))) ## Sequence of exponentially
                                                      ## spaced-apart values
                                                      ## from 0 to 1.
# expo_sd <- sqrt(1 + exp_seq * var_seq[i]) ## Standard deviation used in the
                                            ## normal distribution to produce
                                            ## the data.

## Using this exponentially-spaced sequence of values for the variance,
## the power of the test weakens.



## This code is done by Teng Wei Yeo, 11 Nov 2024.
```

```r
###############################
#### Reproducing Fig. S2. ####
###############################

samples = 100
```

```r
N = 50 # no. of spatially independent locations
days = 91
years = 65


set.seed(1234)

mem_avg = rep(0, samples)

for (i in 1:samples){
  long <- rnorm(years*days*N, 0, 1)
  arr <- array(long, dim = c(years, days, N))
  medians <- apply(arr, MARGIN = c(1,3), median) # years x N matrix
  medians_mat <- medians[, rep(1:N, each = days)] # duplicate median for each day
  medians_arr <- array(medians_mat, dim = c(years,days,N)) # turn back into array
  anomalies <- arr - medians_arr
  max_anoms <- apply(anomalies, MARGIN = c(1,3), max) # years x N matrix
  ranks <- apply(max_anoms, MARGIN = 2, rank) # years x N matrix
  means <- apply(ranks, MARGIN = 1, mean) # 1 x years vector
  year_seq <- seq(1, years, by = 1)
  mem_avg[i] <- coef(lm(means~year_seq))["year_seq"]
}


rejection <- quantile(mem_avg, probs = c(0.025, 0.975))
rejection
```

```
##        2.5%       97.5%
## -0.02793964  0.03402909
```

The rejection region is when the coefficient for year is smaller than -0.0279396 or greater than 0.0340291.

```r
mean_seq <- seq(0,2,0.25)

data_samples <- 100
is_sig_mat <- matrix(0, nrow = data_samples, ncol = length(mean_seq))

for (i in 1:length(mean_seq)){
  print(paste0("Doing ", mean_seq[i], " increase to mean"))
  for (j in 1:data_samples){
    linearised <- seq(0, mean_seq[i], length.out = years)
    linearised_cntr <- scale(linearised, center = TRUE, scale = FALSE)
    data <- rnorm(years*days*N, linearised_cntr, sd = 1)

    arr <- array(data, dim = c(years, days, N))
    medians <- apply(arr, MARGIN = c(1,3), median) # years x N matrix
    medians_mat <- medians[, rep(1:N, each = days)] # same median for each day
    medians_arr <- array(medians_mat, dim = c(years,days,N))
    anomalies <- arr - medians_arr
    max_anoms <- apply(anomalies, MARGIN = c(1,3), max) # years x N matrix
    ranks <- apply(max_anoms, MARGIN = 2, rank) # years x N matrix
    means <- apply(ranks, MARGIN = 1, mean) # average ranks
    year_seq <- seq(1, years, by = 1)
    beta <- coef(lm(means~year_seq))["year_seq"]
```

```r
    is_sig_mat[j,i] = (beta < rejection[1] | beta > rejection[2]) # TRUE/FALSE
  }
}
```

```
## [1] "Doing 0 increase to mean"
## [1] "Doing 0.25 increase to mean"
## [1] "Doing 0.5 increase to mean"
## [1] "Doing 0.75 increase to mean"
## [1] "Doing 1 increase to mean"
## [1] "Doing 1.25 increase to mean"
## [1] "Doing 1.5 increase to mean"
## [1] "Doing 1.75 increase to mean"
## [1] "Doing 2 increase to mean"
```

```r
proportions <- apply(is_sig_mat, MARGIN=2, sum) / data_samples
proportions
```

```
## [1] 0.12 0.10 0.10 0.07 0.10 0.07 0.06 0.08 0.10
```

```r
var_seq <- seq(0, 0.5, 0.05)
is_sig_mat_var <- matrix(0, nrow = data_samples, ncol = length(var_seq))


for (i in 1:length(var_seq)){
  print(paste0("Doing +", var_seq[i]*100, "% change to variance"))
  for (j in 1:data_samples){
    linearised_sd <- sqrt(seq(1, 1 + var_seq[i], length.out = years))
    data <- rnorm(years*days*N, mean = 0, sd = linearised_sd) # recycling

    arr <- array(data, dim = c(years, days, N))
    medians <- apply(arr, MARGIN = c(1,3), median) # years x N matrix
    medians_mat <- medians[, rep(1:N, each = days)] # same median for each day
    medians_arr <- array(medians_mat, dim = c(years,days,N))
    anomalies <- arr - medians_arr
    max_anoms <- apply(anomalies, MARGIN = c(1,3), max) # years x N matrix
    ranks <- apply(max_anoms, MARGIN = 2, rank) # years x N matrix
    means <- apply(ranks, MARGIN = 1, mean)
    year_seq <- seq(1, years, by = 1)
    beta <- coef(lm(means~year_seq))["year_seq"]

    is_sig_mat_var[j,i] = (beta < rejection[1] | beta > rejection[2]) #TRUE/FALSE
  }
}
```

```
## [1] "Doing +0% change to variance"
## [1] "Doing +5% change to variance"
## [1] "Doing +10% change to variance"
## [1] "Doing +15% change to variance"
## [1] "Doing +20% change to variance"
## [1] "Doing +25% change to variance"
## [1] "Doing +30% change to variance"
## [1] "Doing +35% change to variance"
## [1] "Doing +40% change to variance"
## [1] "Doing +45% change to variance"
## [1] "Doing +50% change to variance"
```

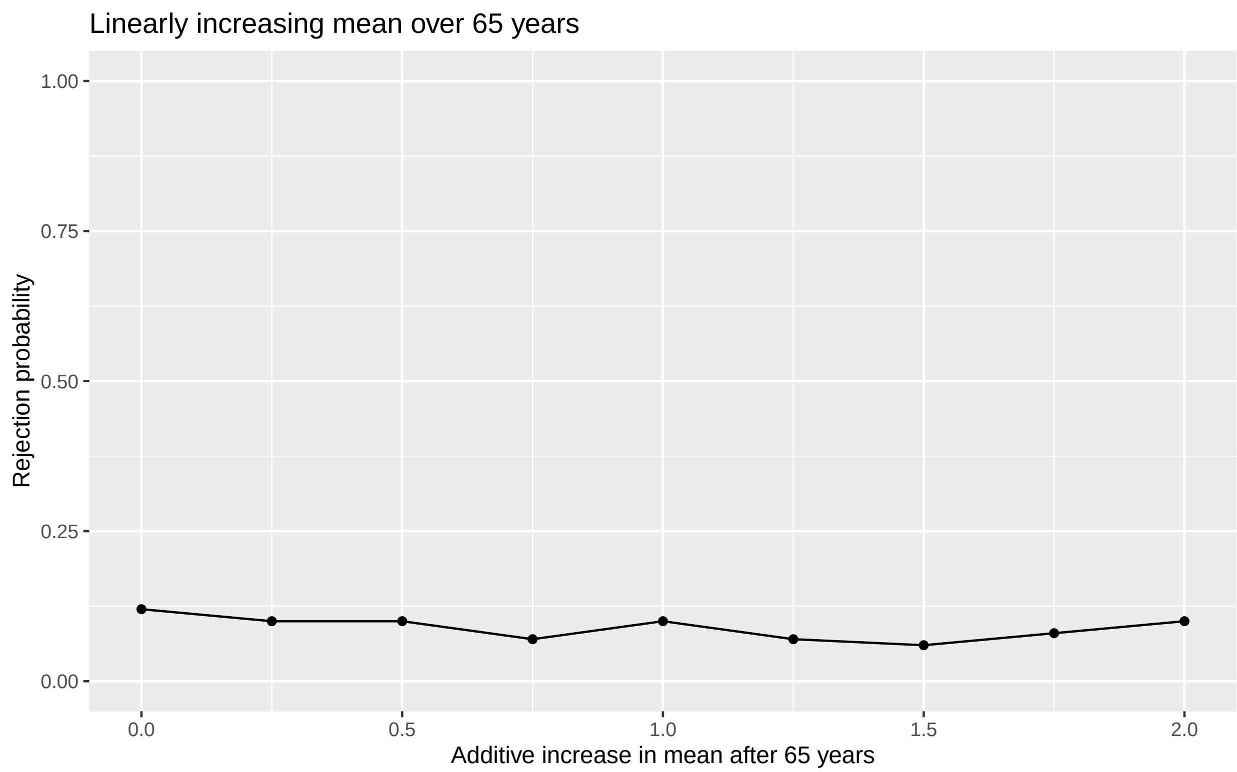
```r
proportions_var <- apply(is_sig_mat_var, MARGIN=2, sum) / data_samples
proportions_var
```

```
##  [1] 0.13 0.51 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
```

```r
## Visualisation

library(ggplot2)
df_means <- data.frame(rej_probab_mean = proportions,
                       mean_change = mean_seq)

ggplot(data = df_means, aes(x = mean_change, y = rej_probab_mean))+
  geom_point()+
  geom_line() +
  labs(y = "Rejection probability",
       x = "Additive increase in mean after 65 years",
       title = "Linearly increasing mean over 65 years") +
  ylim(0, 1)
```

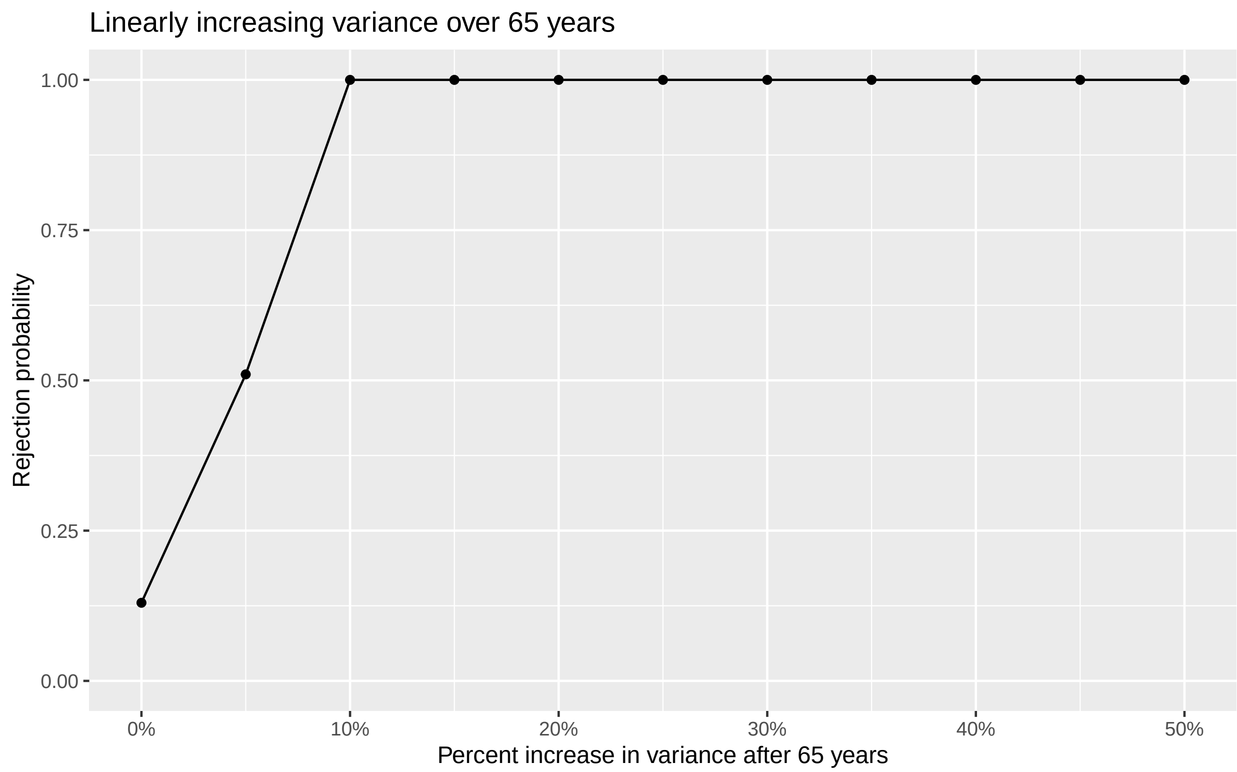


Results are identical to Fig. S2(d).

```r
df_var <- data.frame(rej_probab_var = proportions_var,
                     var_change = var_seq,
                     type = "Linearly increasing")

ggplot(data = df_var, aes(x = var_change, y = rej_probab_var))+
  geom_point()+
  geom_line() +
  labs(y = "Rejection probability",
       x = "Percent increase in variance after 65 years",
```

```
      title = "Linearly increasing variance over 65 years") +
  scale_x_continuous(labels = scales::percent) +
  ylim(0,1)
```



Results are identical to Fig. S2(e).

```
###########################################
#### Quadratically increasing variance ####
###########################################

## This checks whether the test can detect when variance is increasing
## quadratically over time (instead of linearly).

quad_seq <- seq(0, 1, length.out = years)^2

for (i in 1:length(var_seq)){
  print(paste0("Doing +", var_seq[i]*100, "% change to variance"))
  for (j in 1:data_samples){
    quad_sd <- sqrt(1 + quad_seq * var_seq[i])
    data <- rnorm(years*days*N, mean = 0, sd = quad_sd) # recycling

    arr <- array(data, dim = c(years, days, N))
    medians <- apply(arr, MARGIN = c(1,3), median) # years x N matrix
    medians_mat <- medians[, rep(1:N, each = days)] # same median for each day
    medians_arr <- array(medians_mat, dim = c(years,days,N))
    anomalies <- arr - medians_arr
    max_anoms <- apply(anomalies, MARGIN = c(1,3), max) # years x N matrix
    ranks <- apply(max_anoms, MARGIN = 2, rank) # years x N matrix
    means <- apply(ranks, MARGIN = 1, mean)
```

```
    year_seq <- seq(1, years, by = 1)
    beta <- coef(lm(means~year_seq))["year_seq"]

    is_sig_mat_var[j,i] = (beta < rejection[1] | beta > rejection[2])#TRUE/FALSE
  }
}
```

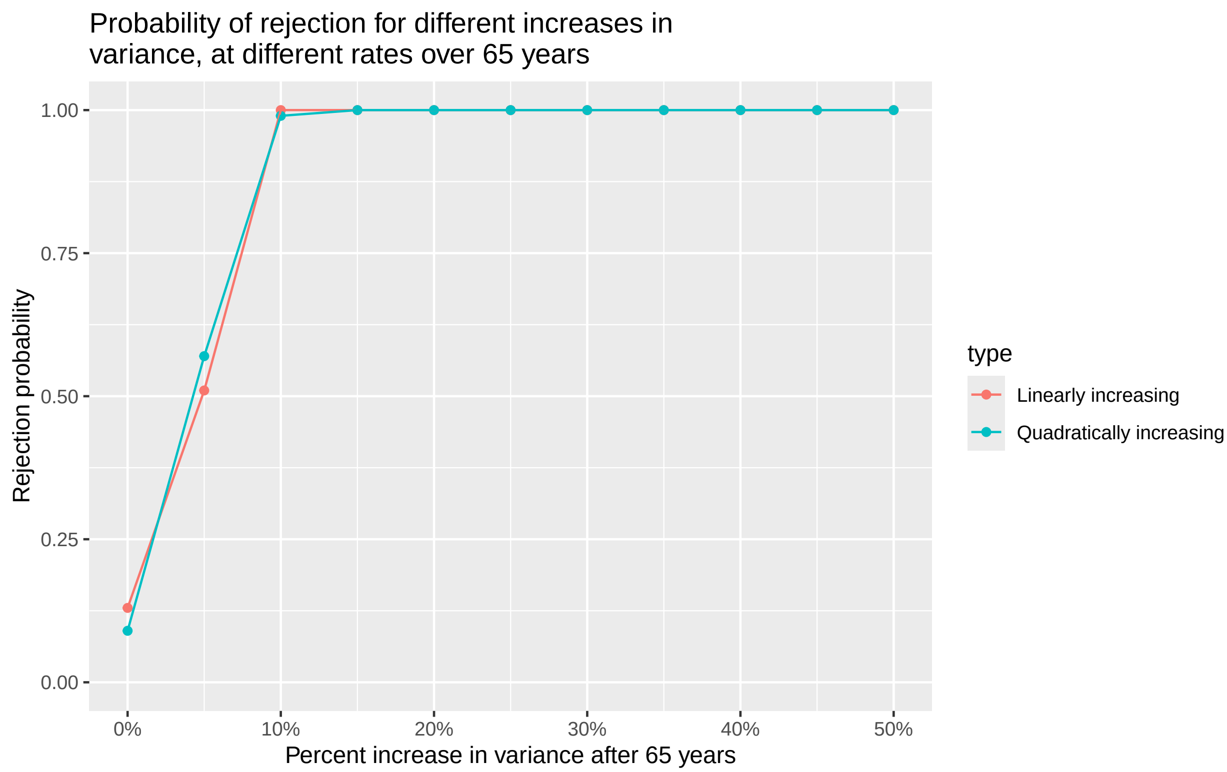
```
## [1] "Doing +0% change to variance"
## [1] "Doing +5% change to variance"
## [1] "Doing +10% change to variance"
## [1] "Doing +15% change to variance"
## [1] "Doing +20% change to variance"
## [1] "Doing +25% change to variance"
## [1] "Doing +30% change to variance"
## [1] "Doing +35% change to variance"
## [1] "Doing +40% change to variance"
## [1] "Doing +45% change to variance"
## [1] "Doing +50% change to variance"
```

```
proportions_var2 <- apply(is_sig_mat_var, MARGIN=2, sum) / data_samples
proportions_var2
```

```
##  [1] 0.09 0.57 0.99 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
```

```
temp <- data.frame(rej_probab_var = proportions_var2,
                   var_change = var_seq,
                   type = "Quadratically increasing")
df_var2 <- rbind(df_var, temp)


ggplot(data = df_var2, aes(x = var_change,
                           y = rej_probab_var,
                           colour = type))+
  geom_point()+
  geom_line() +
  labs(y = "Rejection probability",
       x = "Percent increase in variance after 65 years",
       title = paste0("Probability of rejection for different increases in\n",
                      "variance, at different rates over 65 years")) +
  scale_x_continuous(labels = scales::percent) +
  ylim(0,1)
```



The power of the test remains high for quadratically-increasing variance.

```
############################################
#### Exponentially increasing variance ####
############################################

## This checks whether the test can detect when variance is increasing
## exponentially over time (instead of linearly).
## This is relevant in cases where the variance stays somewhat constant for
## many years, then increases a lot in later years.

## Arbitrary exponentially increasing sequence:
exp_seq <- c(0, exp(1:(years-1)) / exp((years-1)))


for (i in 1:length(var_seq)){
  print(paste0("Doing +", var_seq[i]*100, "% change to variance"))
  for (j in 1:data_samples){
    expo_sd <- sqrt(1 + exp_seq * var_seq[i])
    data <- rnorm(years*days*N, mean = 0, sd = expo_sd) # recycling

    arr <- array(data, dim = c(years, days, N))
    medians <- apply(arr, MARGIN = c(1,3), median) # years x N matrix
    medians_mat <- medians[, rep(1:N, each = days)] # same median for each day
    medians_arr <- array(medians_mat, dim = c(years,days,N))
    anomalies <- arr - medians_arr
    max_anoms <- apply(anomalies, MARGIN = c(1,3), max) # years x N matrix
    ranks <- apply(max_anoms, MARGIN = 2, rank) # years x N matrix
    means <- apply(ranks, MARGIN = 1, mean)
```

```
    year_seq <- seq(1, years, by = 1)
    beta <- coef(lm(means~year_seq))["year_seq"]

    is_sig_mat_var[j,i] = (beta < rejection[1] | beta > rejection[2])#TRUE/FALSE
  }
}
```

```
## [1] "Doing +0% change to variance"
## [1] "Doing +5% change to variance"
## [1] "Doing +10% change to variance"
## [1] "Doing +15% change to variance"
## [1] "Doing +20% change to variance"
## [1] "Doing +25% change to variance"
## [1] "Doing +30% change to variance"
## [1] "Doing +35% change to variance"
## [1] "Doing +40% change to variance"
## [1] "Doing +45% change to variance"
## [1] "Doing +50% change to variance"
```

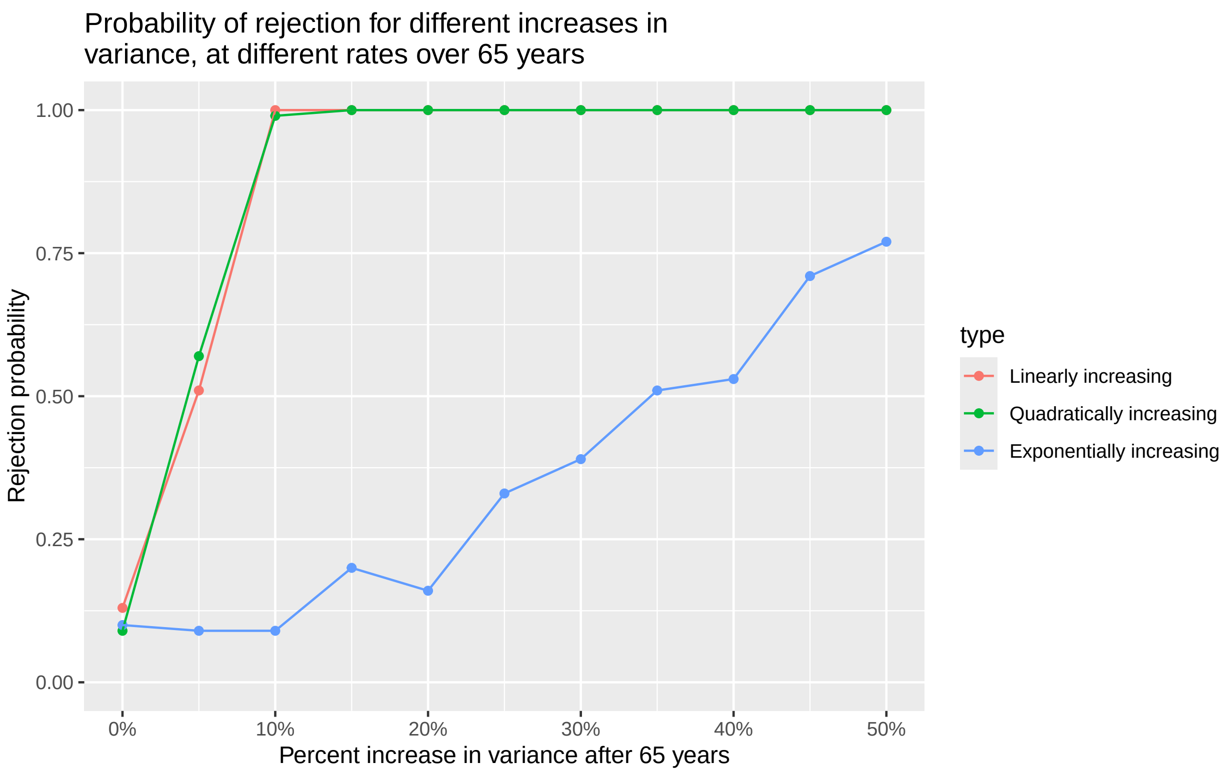
```
proportions_var3 <- apply(is_sig_mat_var, MARGIN=2, sum) / data_samples
proportions_var3
```

```
##  [1] 0.10 0.09 0.09 0.20 0.16 0.33 0.39 0.51 0.53 0.71 0.77
```

```
temp2 <- data.frame(rej_probab_var = proportions_var3,
                   var_change = var_seq,
                   type = "Exponentially increasing")
df_var3 <- rbind(df_var2, temp2)

df_var3$type <-
  factor(df_var3$type, levels = c("Linearly increasing",
                                  "Quadratically increasing",
                                  "Exponentially increasing"))

ggplot(data = df_var3, aes(x = var_change,
                           y = rej_probab_var,
                           colour = type))+
  geom_point()+
  geom_line() +
  labs(y = "Rejection probability",
       x = "Percent increase in variance after 65 years",
       title = paste0("Probability of rejection for different increases in\n",
                      "variance, at different rates over 65 years")) +
  scale_x_continuous(labels = scales::percent) +
  ylim(0,1)
```



The test is weaker for the exponential case.

This is in part due to the nature of the test in being successful at accounting for the presence of a change in variance, but not for accounting the magnitude of the change in variance.

Since the 'ramp' of exp() is quite steep (i.e., it takes until the 59th year for exp_seq > 0.001), there is only a small change in variance for most of the years. The variance increases by a lot only in the last few years:

```
exp_seq
```

```
##  [1] 0.000000e+00 4.359610e-28 1.185065e-27 3.221340e-27 8.756511e-27
##  [6] 2.380266e-26 6.470235e-26 1.758792e-25 4.780893e-25 1.299581e-24
## [11] 3.532629e-24 9.602680e-24 2.610279e-23 7.095474e-23 1.928750e-22
## [16] 5.242886e-22 1.425164e-21 3.873998e-21 1.053062e-20 2.862519e-20
## [21] 7.781132e-20 2.115131e-19 5.749522e-19 1.562882e-18 4.248354e-18
## [26] 1.154822e-17 3.139133e-17 8.533048e-17 2.319523e-16 6.305117e-16
## [31] 1.713908e-15 4.658886e-15 1.266417e-14 3.442477e-14 9.357623e-14
## [36] 2.543666e-13 6.914400e-13 1.879529e-12 5.109089e-12 1.388794e-11
## [41] 3.775135e-11 1.026188e-10 2.789468e-10 7.582560e-10 2.061154e-09
## [46] 5.602796e-09 1.522998e-08 4.139938e-08 1.125352e-07 3.059023e-07
## [51] 8.315287e-07 2.260329e-06 6.144212e-06 1.670170e-05 4.539993e-05
## [56] 1.234098e-04 3.354626e-04 9.118820e-04 2.478752e-03 6.737947e-03
## [61] 1.831564e-02 4.978707e-02 1.353353e-01 3.678794e-01 1.000000e+00
```

— END OF DOCUMENT —