Homework 2 Solution

Due: 2/18/2019 before 11pm. Submit Rmd file and the html output file in Canvas (file upload).

1. [30 points]

Does normal distribution assumption matter?

To try to answer this question, we run a simulation study in the context of regression. Consider

$$y_i = x_i'\beta + \epsilon_i$$

for $i = 1, \ldots, n$, where $x_i = (x_{i1}, \ldots, x_{ip})'$ are the covariates, and $\beta = (\beta_1, \ldots, \beta_p)'$ are the regression coefficients. Calculate the least square estimator $\widehat{\beta} = (\widehat{\beta}_1, \ldots, \widehat{\beta}_p)'$ for β, and their 95% confidence intervals based on normal approximation. Calculate the empirical coverage for each of the coefficient, denoted as p_j, over 1000 repititions. Separatly report the average empirical coverages for the nonzero regression coefficients (average p_j over nonzero β_j) and the zero coefficients (average p_j over zero β_j). Consider the combination of the following settings:

a. $n = 40$, $n = 80$, comment on the effect of sample size;

b. The first half of β are 1 and the rest dimensions are zero. Consider $p = 10$ and $p = 30$. Comment on the effect of dimension;

c. Generate the covariate x_i from multivariate normal distribution with mean 0 and covariance $\Sigma = (\sigma_{j_1 j_2})$ for $\sigma_{j_1 j_2} = 0.75^{|j_1 - j_2|}$. This is the AR covariance structrue for the covariates;

d. Generate the error ϵ_i from the following scenarios, and comment on the imapct of the variance distribution on the empirical covarages.

- N(0, 1)
- standardized t distribution with 3 degree of freedom (standardized so that it has variance 1)
- standardized t distribution with 10 degree of freedom (standardized so that it has variance 1)
- exponential distribution with parameter 1 (standardized so that it has mean 0)

Solution:

```
library(tidyverse)
library(kableExtra)

GetCoverage <- function(n, p, N = 1e3) {
  beta <- rep(1:0, each = p / 2)
  Sigma <- .75 ^ abs(outer(1:p, 1:p, "-"))

  re <- replicate(N, {
    x <- MASS::mvrnorm(n, mu = rep(0, p), Sigma = Sigma)
    e_list <- list(normal = rnorm(n), t3 = rt(n, 3) / sqrt(3),
                   t10 = rt(n, 10) / sqrt(1.25), exp = rexp(n) - 1)
    ey <- x %*% beta

    sapply(e_list, function(e) {
      y <- ey + e
```

Table 1: Coverage Rate for β

		$N(0,1)$		t_3		t_{10}		$exp(1)$	
n	p	$\beta \neq 0$	$\beta = 0$	$\beta \neq 0$	$\beta = 0$	$\beta \neq 0$	$\beta = 0$	$\beta \neq 0$	$\beta = 0$
40	10	0.9432	0.9372	0.9394	0.9380	0.9394	0.9372	0.9378	0.9410
40	30	0.9169	0.9203	0.9213	0.9227	0.9174	0.9215	0.9209	0.9253
80	10	0.9476	0.9438	0.9452	0.9440	0.9418	0.9444	0.9486	0.9380
80	30	0.9417	0.9413	0.9440	0.9409	0.9474	0.9447	0.9443	0.9449

```
      ci <- confint.default(lm(y ~ x + 0))
      beta >= ci[, 1] & beta <= ci[, 2]
    })
  })

  coverage <- rowMeans(re, dims = 2)
  sapply(c(nonzero = 1, zero = 0), function(x) colMeans(coverage[beta == x, ]))
}


set.seed(1)
arg <- expand.grid(p = c(10, 30), n = c(40, 80))
res <- mapply(GetCoverage, n = arg$n, p = arg$p, N = 1e3, SIMPLIFY = F)
tab <- sapply(res, function(x) c(t(x))) %>% t() %>% cbind(arg[, 2:1], .)
kable(tab, "latex", digits = 4, escape = F, booktabs = T, caption = "Coverage Rate for $\\beta$",
      col.names = c("$n$", "$p$", rep(c("$\\beta\\neq0$", "$\\beta=0$"), 4))) %>%
  kable_styling() %>%
  add_header_above(c(" " = 2, "$N(0,1)$" = 2, "$t_3$" = 2, "$t_{10}$" = 2, "$exp(1)$" = 2), escape = F)
```

From the table,

- As n increases, the coverage rate increases and gets close to 95%.
- As p increases, the coverage rate decreases and gets away from 95%.
- When n is small and p is large, the coverage rate is slow, only 92%.
- When n is large and p is small, the coverage rate reaches at least 94.6%.
- There is no difference in coverage rate between non-zero β and zero β.
- 4 scenarios of ϵ: all these scenarios perform well, and there is no much difference for all cases except for $n = 40, p = 30$. For the case $n = 40, p = 30$ (when p is about as large as n):
  - $N(0,1)$: has the highest coverage rate, which is closest to 95%. Performs very well.
  - t_3: a bit lower coverage rate than $N(0,1)$. Performs well.
  - t_{10}: very similar as $N(0,1)$. Performs very well.
  - exp(1): has the lowest coverage rate. Performs well when n is large.

2. [30 points]

Let $X_1, \ldots, X_n$ be i.i.d. **univariate** random variables from $N(\mu, \sigma^2)$. Answer the following questions:

a. Provide the conditional distribution of $X_1, \ldots, X_{n-1}$ given $\bar{X}$. [10 points]

Solution: Let $Y = (X_1, \ldots, X_n, \bar{X})'$, then

$$Y = AX = \begin{pmatrix} I_n \\ \frac{1}{n}\mathbf{1}_n' \end{pmatrix} X \sim N(A\boldsymbol{\mu}, \sigma^2 AA') = N\left(\boldsymbol{\mu}_{n+1}, \sigma^2 \begin{pmatrix} I_n & \frac{1}{n}\mathbf{1}_n \\ \frac{1}{n}\mathbf{1}_n' & \frac{1}{n} \end{pmatrix}\right)$$

By conditional distribution formula,

$$X|\bar{X} \sim N(\mu_{1|2}, \sigma^2\Sigma_{1|2})$$

where

$$\mu_{1|2} = \boldsymbol{\mu}_n + \frac{1}{n}\mathbf{1}_n \cdot n(\bar{X} - \mu) = \bar{X}\mathbf{1}_n$$
$$\Sigma_{1|2} = I_n - \frac{1}{n}\mathbf{1}_n \cdot n \cdot \frac{1}{n}\mathbf{1}_n' = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n'$$

Thus,

$$(X_1, \ldots, X_{n-1})'|\bar{X} \sim N(\bar{X}\mathbf{1}_{n-1}, \sigma^2 I_{n-1} - \sigma^2/n \cdot \mathbf{1}_{n-1}\mathbf{1}_{n-1}')$$

b. Write a function in R which will generate a random vector $X = (X_1, \ldots, X_{n-1}, X_n)$ given $\bar{X}$. (Note that $X_n = n\bar{X} - \sum_{i=1}^{n-1} X_i$.) [10 points]

```
recover <- function(n, xbar, sigma2) {
  MASS::mvrnorm(1, mu = rep(xbar, n), Sigma = (diag(n) - 1 / n) * sigma2)
  # another faster solution
  # x <- rnorm(n, 0, sqrt(sigma2))
  # x - mean(x) + xbar
}
```

c. Suppose we have a 16×16 matrix A which is averaged in each of the 4×4 non-overlapping blocks, resulting a 4×4 matrix B. Based on the result from (b), write a function to predict the original matrix A. Try your function on a setting of specific B and σ^2. [10 points]

```
B <- matrix(1:16, 4)
A <- lapply(1:4, function(i) {
  lapply(1:4, function(j) matrix(recover(16, xbar = B[i, j], sigma2 = 1), 4)) %>%
do.call(cbind, .)
}) %>% do.call(rbind, .)
options(width = 100)
B; round(A, 1)
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
```

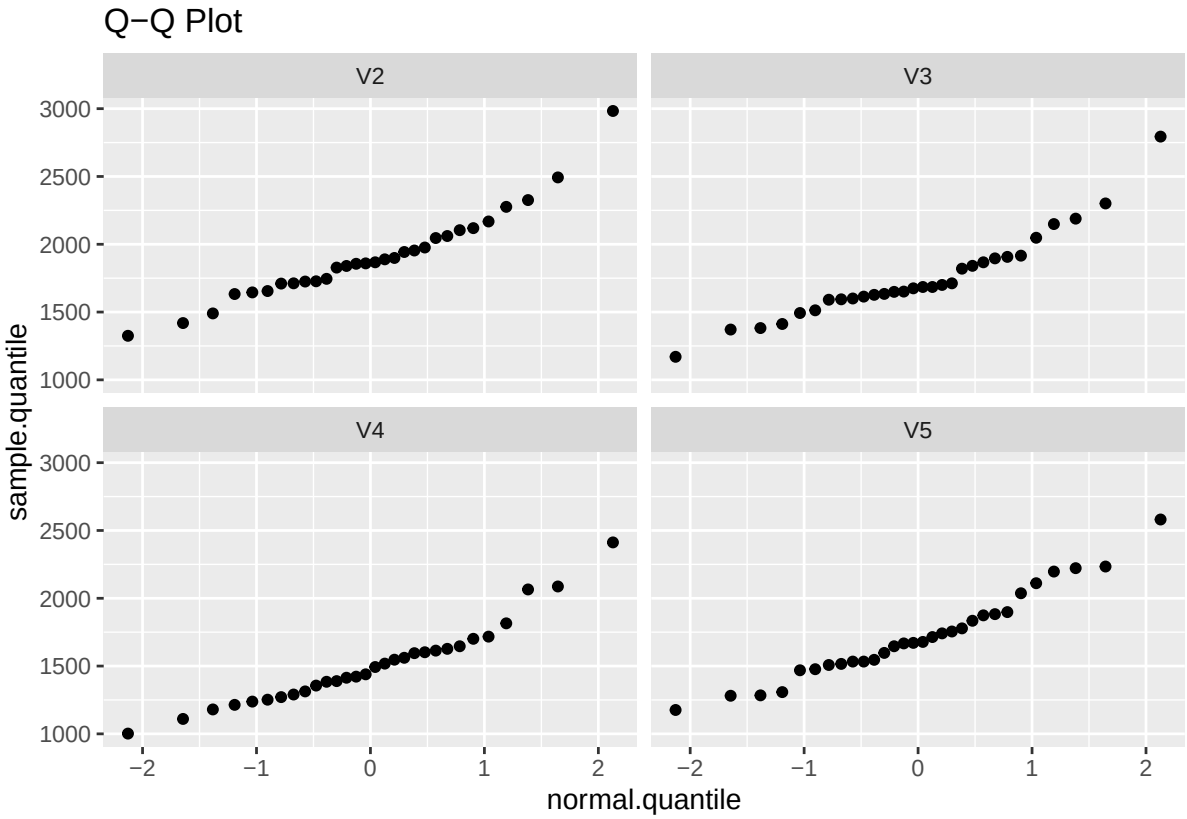
```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16]
 [1,]  1.7  0.3 -0.7  1.5  5.9  4.9  4.1  3.9  9.0   9.2   9.6   8.7  12.7  14.1  12.7  13.4
 [2,]  0.6  1.6  2.7  0.1  6.6  3.6  4.3  6.1  9.2   7.2   8.7   7.8  13.2  13.6  11.7  12.9
 [3,]  0.3  1.6 -0.1  2.6  5.7  5.3  5.6  6.6  9.8  10.2   9.3   8.7  15.3  12.9  11.7  12.0
 [4,] -0.7  0.8  1.9  1.9  4.4  5.6  3.9  3.5 10.3   9.6   7.1   9.7  13.7  11.4  12.1  14.6
 [5,]  3.2  2.2  3.4  2.8  5.0  5.1  6.0  5.9 11.7  10.8  10.8  10.6  16.3  12.3  13.7  12.6
 [6,]  2.2  1.5  2.8  2.3  6.8  6.6  7.0  7.0  9.2   9.9   9.2  10.5  13.8  13.0  13.8  13.5
 [7,]  1.6  1.2  2.3  1.2  4.4  5.0  6.5  6.0 10.8  10.6   8.7  11.3  14.7  14.3  16.3  14.2
 [8,] -0.2  0.9  3.8  0.9  7.0  4.5  6.1  7.2 11.8   8.8   6.7   8.5  14.2  13.8  13.2  14.4
 [9,]  2.9  1.3  1.4  2.1  7.1  6.3  6.0  9.7 10.6  11.7  10.8   9.4  16.1  15.1  14.6  14.6
[10,]  2.4  2.4  0.5  3.2  6.9  6.7  6.9  5.5 11.3  11.1  10.9   8.9  14.7  14.2  16.3  17.2
[11,]  4.3  3.3  3.9  4.6  8.4  8.6  6.0  5.9 11.4  11.4  13.2  11.2  14.6  14.4  15.0  14.1
[12,]  4.2  4.7  4.1  2.8  5.9  8.0  5.9  8.3 10.4  10.7  11.5  11.5  16.1  12.7  13.9  16.4
[13,]  4.1  3.3  4.6  3.2  9.0  8.6  7.3  6.9 10.9  11.3  11.1  11.8  15.0  14.8  16.1  16.2
[14,]  4.0  4.6  4.7  5.1  7.0  7.2  8.2  7.7 13.1  13.5  11.5  13.5  17.5  17.2  16.7  15.1
[15,]  3.9  4.7  4.6  3.3  7.9 10.4  8.0  9.4 13.6  11.8  11.3  12.3  15.4  16.1  15.7  15.8
[16,]  3.0  2.6  4.7  3.7  7.4  7.2  7.9  8.1 10.6  11.7  11.3  12.7  16.5  15.6  17.0  15.2
```

3. [40 points]

Use the data `board.stiffness.dat` (the stiffness at four locations of a board) and code `testnormality.R` provided in the folder `Data and code` in Canvas.

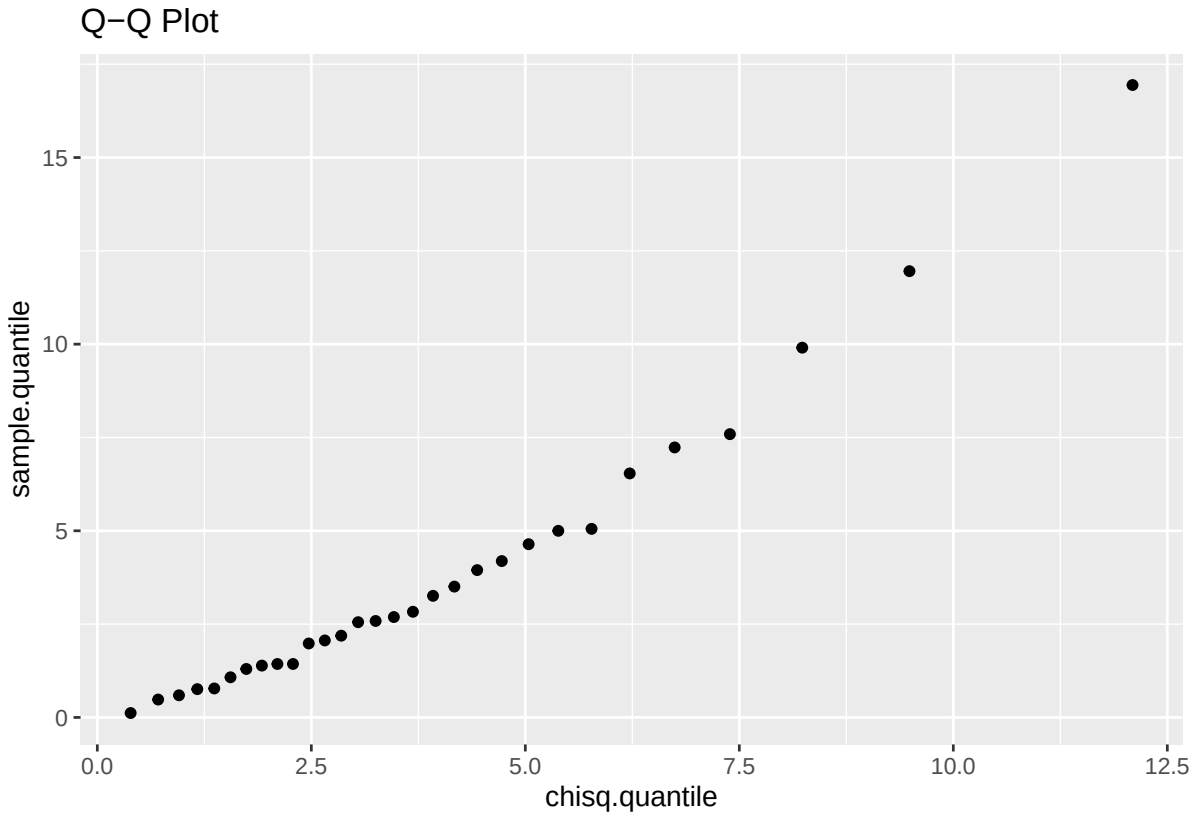
a. Use `QQplot.normal` function in the code and `ggplot` to make quantile plots for each of the four variables. Use `facet` option to put them in one plot. Comment. [10 points]

```
source("testnormality.R")

X[, -1] %>% reshape2::melt() %>% group_by(variable) %>%
  summarise(x = list(QQplot.normal(value))) %>% unnest(cols = c(x)) %>%
  qplot(x = normal.quantile, y = sample.quantile, data = ., facets = ~ variable, main = "Q-Q Plot")
```



b. Write a function to conduct Chi-square quantile plot. Implement this function on this dataset. [10 points]

```
QQplot.chisq <- function(x) {
  if (is.vector(x)) x <- matrix(x)
  d.sort <- sort(mahalanobis(x, colMeans(x), var(x)))
  n <- NROW(x); p <- NCOL(x)
  q <- qchisq((1:n - .5) / n, df = p)
  data.frame(sample.quantile = d.sort, chisq.quantile = q)
}


X[, -1] %>% QQplot.chisq() %>%
  qplot(x = chisq.quantile, y = sample.quantile, data = ., main = "Q-Q Plot")
```



c. Based on the fucntion `testnormality` in the code, conduct random projection based shapiro-wilks test for multivariate data. Change the code to return the minimum pvalue of the shapiro-wilks test among all the random projections. (The original code returns the qvalue, another statistic for the test). Implement this function on the stiffness data. [10 points]

```
testnormality.min <- function(X, numproj = 1e3) {
  p <- NCOL(X)
  x <- matrix(rnorm(numproj * p), p)
  z <- x / rep(sqrt(colSums(x ^ 2)), rep(p, numproj))
  tempdat <- as.matrix(X) %*% z
  pvals <- apply(tempdat, 2, function(x) shapiro.test(x)$p.value)
  min(pvals)
}


p_min <- testnormality.min(X[, -1]); p_min
```

```
## [1] 1.03023e-05
```

d. Use parametric bootstrap to determine the reject region of the test. Shall we reject the multivariate normality hypotheses on the stiffness data? [10 points]

```
bootstrap <- function(X, B, numproj = 1e3) {
  n <- NROW(X); p <- NCOL(X)
  sample <- MASS::mvrnorm(n = n * B, mu = colSums(X), Sigma = var(X))
  dim(sample) <- c(n, p, B)
  apply(sample, 3, testnormality.min, numproj = numproj)
}
```

```r
p_min_sample <- bootstrap(X[, -1], B = 1e3)
quantile(p_min_sample, c(.025, .975))
```

```
##         2.5%        97.5%
## 0.0004497442 0.0355610262
```

Using bootstrap sample size $B = 1000$, we can see that the min p-value of data (1.0302299×10^{-5}) is out of the 95% bootstrap interval. Thus, we reject the multivariate normality hypotheses on the stiffness data.