STAT 824 hw 05

Influence functions, Edgeworth expansions, bootstrap, residual and wild bootstrap in regression

1. Consider the functional $\sigma^2 = T(F) = \int (x - \int t dF(t))^2 dF(x)$.

   (a) Find the influence function φ_F.

Solution: First we have

$$\begin{aligned}
T(F + \varepsilon(\delta_x - F)) &= \int (s - \int t d(F + \varepsilon(\delta_x - F))(t))^2 d(F + \varepsilon(\delta_x - F))(s) \\
&= \int (s - (\mu + \varepsilon(x - \mu)))^2 d(F + \varepsilon(\delta_x - F))(s) \\
&= \int ((s - \mu) - \varepsilon(x - \mu))^2 d(F + \varepsilon(\delta_x - F))(s) \\
&= \int ((s - \mu)^2 - 2\varepsilon(s - \mu)(x - \mu) + \varepsilon^2(x - \mu)^2) d((1 - \varepsilon)F + \varepsilon\delta_x)(s) \\
&= (1 - \varepsilon)\sigma^2 + (1 - \varepsilon)\varepsilon^2(x - \mu)^2 + \varepsilon(x - \mu)^2 - 2\varepsilon^2(x - \mu)^2 + \varepsilon^2(x - \mu)^2 \\
&= (1 - \varepsilon)\sigma^2 + \varepsilon^3(x - \mu)^2 + \varepsilon(x - \mu)^2.
\end{aligned}$$

Note we have written $\mu = \int x dF(x)$. The influence function is given by

$$\varphi_F(x) = \frac{d}{d\varepsilon}\left[(1 - \varepsilon)\sigma^2 + \varepsilon^3(x - \mu)^2 + \varepsilon(x - \mu)^2\right]\Big|_{\varepsilon=0} = (x - \mu)^2 - \sigma^2.$$

   (b) Identify the remainder R_n in the expansion $\sqrt{n}(\hat{\sigma}_n^2 - \sigma^2) = n^{-1/2}\sum_{i=1}^n \varphi_F(X_i) + \sqrt{n}R(\hat{F}_n - F)$ and argue that it converges in probability to 0 as $n \to \infty$.

Solution: Note that the plug-in estimator of σ^2 is $\hat{\sigma}_n^2 = T(\hat{F}_n) = n^{-1}\sum_{i=1}^n (X_i - \bar{X}_n)^2$. The von Mises expansion is

$$\begin{aligned}
\sqrt{n}(\hat{\sigma}_n^2 - \sigma^2) &= \frac{1}{\sqrt{n}}\sum_{i=1}^n ((X_i - \mu)^2 - \sigma^2) + \sqrt{n}R(\hat{F}_n - F) \\
&= \frac{1}{\sqrt{n}}\sum_{i=1}^n ((X_i - \bar{X}_n) + (\bar{X}_n - \mu))^2 - \sigma^2) + \sqrt{n}R(\hat{F}_n - F) \\
&= \sqrt{n}\frac{1}{n}\sum_{i=1}^n (X_i - \bar{X}_n)^2 - \sqrt{n}\sigma^2 + \sqrt{n}(\bar{X}_n - \mu)^2 + \sqrt{n}R(\hat{F}_n - F) \\
&= \sqrt{n}(\hat{\sigma}_n^2 - \sigma^2) + \sqrt{n}(\bar{X}_n - \mu)^2 + \sqrt{n}R(\hat{F}_n - F),
\end{aligned}$$

which gives

$$\sqrt{n}R(\hat{F}_n - F) = -\sqrt{n}(\bar{X}_n - \mu)^2.$$

We can write the remainder as the product of a sequence of random variables converging in distribution to the Normal$(0, \sigma^2)$ distribution and a sequence of random variables converging to 0 in probability, so that $\sqrt{n}R(\hat{F}_n - F) \xrightarrow{\text{p}} 0$ as $n \to \infty$. We write

$$-\sqrt{n}(\bar{X}_n - \mu)^2 = \underbrace{-\sqrt{n}(\bar{X}_n - \mu)}_{\xrightarrow{\text{D}}\text{Normal}(0,\sigma^2)}\underbrace{(\bar{X}_n - \mu)}_{\xrightarrow{\text{p}}0} \xrightarrow{\text{p}} 0 \text{ as } n \to \infty.$$

(c) Give the asymptotic behavior of $\sqrt{n}(\hat{\sigma}_n^2 - \sigma^2)$ as $n \to \infty$.

Solution: From the von Mises expansion, we see that $\sqrt{n}(\hat{\sigma}_n^2 - \sigma^2)$ has the same limiting distribution as $n^{-1/2}\sum_{i=1}^n((X_i - \mu)^2 - \sigma^2)$. The central limit theorem gives

$$n^{-1/2}\sum_{i=1}^{n}((X_i - \mu)^2 - \sigma^2) \xrightarrow{\text{D}} \text{Normal}(0, \vartheta),$$

where $\vartheta = \text{Var}((X_1 - \mu)^2) = \mathbb{E}(X_1 - \mu)^4 - (\mathbb{E}(X_1 - \mu)^2)^2 = \mu_4 - \sigma^4$, where μ_4 is the fourth central moment of X_1. So we have

$$\sqrt{n}(\hat{\sigma}_n^2 - \sigma^2) \xrightarrow{\text{D}} \text{Normal}(0, \mu_4 - \sigma^4).$$

2. (a) For a Normal random variable:
   i. Give μ_3, the third central moment.

   Solution: The third central moment of every symmetric distribution is equal to 0.

   ii. Give μ_4/σ^4, where μ_4 is the fourth central moment and σ^2 is the variance.

   Solution: Differentiating $e^{\sigma^2 t^2/2}$ four times with respect to t and evaluating this at $t = 0$ gives $\mu_4 = 3\sigma^4$, so that $\mu_4/\sigma^4 = 3$.

(b) Suppose $X_1, \ldots, X_n \overset{\text{ind}}{\sim} F$ with $\mathbb{E}X_1 = \mu$ and $\text{Var}\, X_1 = \sigma^2 < \infty$.
   i. Explain why we expect fast convergence to Normality of $\sqrt{n}(\bar{X}_n - \mu)/\sigma$ when F is symmetric.

   Solution: The 1st-order Edgeworth expansion for the cdf of $\sqrt{n}(\bar{X}_n - \mu)/\sigma$ is given by

   $$\Psi_{n,3}(x) = \Phi(x) - \frac{1}{6\sqrt{n}}\frac{\mu_3}{\sigma^3}(x^2 - 1)\phi(x).$$

   The second term will be equal to zero if $\mu_3 = 0$ so that

   $$\sup_{x \in \mathbb{R}} \left| P(\sqrt{n}(\bar{X}_n - \mu)/\sigma) - \Phi(x) \right| = o(n^{-1/2}).$$

   For a distribution which is not symmetric, the distance is of order $O(n^{-1/2})$.

   ii. Suppose F has the same skewness (μ_3/σ^3) and kurtosis (μ_4/σ^4) as the Normal distribution. Explain why we expect the convergence to Normality of $\sqrt{n}(\bar{X}_n - \mu)/\sigma$ to be super fast.

Solution: The 2nd-order Edgeworth expansion for the cdf of $\sqrt{n}(\bar{X}_n - \mu)/\sigma$ is given by

$$\Psi_{n,4}(x) = \Phi(x) - \frac{1}{6\sqrt{n}}\frac{\mu_3}{\sigma^3}(x^2-1)\phi(x) - \frac{1}{24n}(\frac{\mu_4}{\sigma^4} - 3)(x^3 - 3x) - \frac{1}{72n}\frac{\mu_3^2}{\sigma^6}(x^5 - 10x^3 + 15x).$$

All terms apart from $\Phi(x)$ will disappear from the right hand side if $\mu_3 = 0$ and $\mu_4/\sigma^4 = 3$, so that

$$\sup_{x\in\mathbb{R}} \Big| P(\sqrt{n}(\bar{X}_n - \mu)/\sigma) - \Phi(x) \Big| = o(n^{-1}).$$

Convergence to Normality will in this case be very fast.

3. Let $X_1, \ldots, X_n \overset{\text{ind}}{\sim} F$ with $\operatorname{Var} X_1 < \infty$. Let T be the statistical functional $T(F) = g(\int x dF(x))$, for a differentiable function $g : \mathbb{R} \to \mathbb{R}$ with derivative g' satisfying $|g'(x) - g'(x')| \leq L|x - x'|$ for all $x, x' \in \mathbb{R}$. In the von Mises expansion

$$\sqrt{n}(T(\hat{F}_n) - T(F)) = \frac{1}{\sqrt{n}}\sum_{i=1}^{n} \varphi_F(X_i) + \sqrt{n}R(\hat{F}_n - F),$$

give a detailed argument for why $\sqrt{n}R(\hat{F}_n - F)$ becomes negligible.

Solution: We have from the course notes that the expansion can be written as

$$\sqrt{n}(g(\bar{X}_n) - g(\mu)) = g'(\mu)\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(X_i - \mu) + \sqrt{n}R(\hat{F}_n - F)$$

By Taylor's theorem we may write

$$\sqrt{n}(g(\bar{X}_n) - g(\mu)) = \sqrt{n}g'(\mu + \tau(\bar{X}_n - \mu))(\bar{X}_n - \mu) \quad \text{for some } \tau \in (0,1),$$

so we have

$$\begin{aligned}
\sqrt{n}R(\hat{F}_n - F) &= \sqrt{n}[g'(\mu + \tau(\bar{X}_n - \mu)) - g'(\mu)](\bar{X}_n - \mu) \\
&\leq |\bar{X}_n - \mu| \cdot \sqrt{n}(\bar{X}_n - \mu) \\
&\leq \frac{1}{\sqrt{n}}[\sqrt{n}(\bar{X}_n - \mu)]^2 \\
&= \frac{\sigma^2}{\sqrt{n}}[\sqrt{n}(\bar{X}_n - \mu)/\sigma]^2 \qquad \big(\sigma^2 = \operatorname{Var} X_1\big) \\
&\to 0 \text{ in probability as } n \to \infty
\end{aligned}$$

since $[\sqrt{n}(\bar{X}_n - \mu)/\sigma]^2$ converges in distribution to a χ^2_1 random variable.

4. Let $X_1, \ldots, X_n \overset{\text{ind}}{\sim} F$ with density f. Let $\xi_{\tau_1}, \xi_{\tau_2}$ be the τ_1, τ_2 quantiles of F, with $0 < \tau_1 < \tau_2 < 1$.

(a) Use von Mises expansions to find ϑ such that

$$\sqrt{n}[(\hat{\xi}_{\tau_2} - \hat{\xi}_{\tau_1}) - (\xi_{\tau_2} - \xi_{\tau_1})] \to \text{Normal}(0, \vartheta)$$

in distribution as $n \to \infty$.

Solution: We begin by writing

$$\xi_{\tau_2} - \xi_{\tau_1} = T(F) = F^{-1}(\tau_2) - F^{-1}(\tau_1).$$

Then, using the form of the influence curve for each quantile, we may write

$$\sqrt{n}(T(\hat{F}_n) - T(F)) \approx \sqrt{n} \cdot \frac{1}{n} \sum_{i=1}^{n} \left[\frac{\tau_2 - \mathbf{1}(X_i \leq \xi_{\tau_2})}{f(\xi_{\tau_2})} - \frac{\tau_1 - \mathbf{1}(X_i \leq \xi_{\tau_1})}{f(\xi_{\tau_1})} \right],$$

assuming the remainder term to be small. The right hand side of the above is asymptotically Normal with variance

$$\vartheta = \text{Var}\left[\varphi_F(X_1)\right] = \frac{\tau_1(1-\tau_1)}{[f(\xi_{\tau_1})]^2} + \frac{\tau_2(1-\tau_2)}{[f(\xi_{\tau_2})]^2} - 2\frac{\tau_1(1-\tau_2)}{f(\xi_{\tau_1})f(\xi_{\tau_2})}.$$

(b) Note that evaluations of the density f appear in the expression for ϑ. Consider replacing f with a kernel density estimate $\hat{f}_n$ (under some choice of bandwidth h) and replacing $F^{-1}(\tau_1)$ and $F^{-1}(\tau_2)$ with $X_{(\lceil \tau_1 n \rceil)}$ and $X_{(\lceil \tau_2 n \rceil)}$, respectively, thereby constructing an estimator $\hat{\vartheta}$ of ϑ. Give the form of an asymptotic $(1-\alpha)100\%$ CI for $\xi_{\tau_2} - \xi_{\tau_1}$, assuming that your estimator $\hat{\vartheta}$ is a consistent estimator of ϑ.

Solution: An asymptotic $(1-\alpha)100\%$ CI for $\xi_{\tau_2} - \xi_{\tau_1}$ is given by

$$\hat{\xi}_{\tau_2} - \hat{\xi}_{\tau_1} \pm z_{\alpha/2}\sqrt{\hat{\vartheta}}/\sqrt{n},$$

provided $\hat{\vartheta}$ is a consistent estimator for ϑ.

(c) Choose a distribution F which is highly non-Normal. Then, for some small sample size n (say between 20 and 50), draw 500 data sets, and with each data set construct the confidence interval you described in part (b) for the IQR (the 0.75 quantile minus the 0.25 quantile). Specifically:

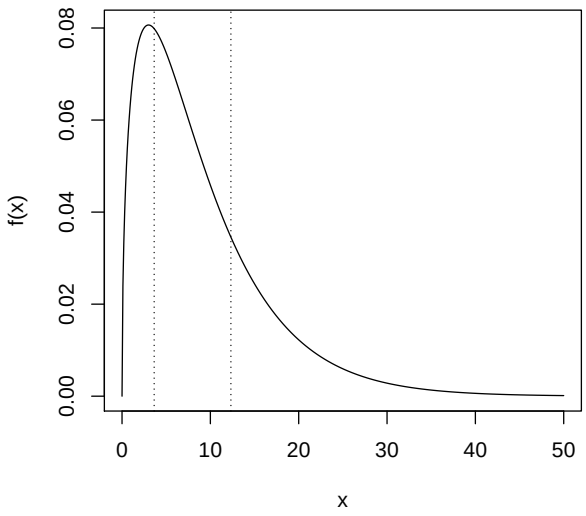
i. Make a plot showing the true density f corresponding to your distribution F. Indicate the 0.25 and 0.75 quantiles.

ii. Report the proportion of times the confidence interval contained the true IQR.

iii. Include your R code.

Note that you must choose a bandwidth for estimating f. You may simply use the `density()` *function in R to do this if you wish, which has a default way of selecting a bandwidth.*

Solution: I generated data from the Gamma$(1.5, 6)$ distribution, of which the pdf looks like this, with IQR indicated:



Here is my code:

```
alpha <- 1.5
beta <- 6
tau1 <- .25
tau2 <- .75

xi1 <- qgamma(tau1,alpha,scale=beta)
xi2 <- qgamma(tau2,alpha,scale=beta)
dxi <- xi2 - xi1

n <- 20
S <- 500
sqrt_n <- sqrt(n)
z_val <- qnorm(.975)
asymp.v.hat.lower <- asymp.v.hat.upper <- numeric(S)
for( s in 1:S ){

  X <- rgamma(n,alpha,scale = beta)

  xi1.hat <- quantile(X,probs=tau1)
  xi2.hat <- quantile(X,probs=tau2)
  dxi.hat <- xi2.hat - xi1.hat

  h <- density(x = X)$bw
  f.hat.xi1.hat <- mean(dnorm((X - xi1.hat)/h))/h
```

```
  f.hat.xi2.hat <- mean(dnorm((X - xi2.hat)/h))/h

  v1.hat <- tau1*(1-tau1) / f.hat.xi1.hat^2
  v2.hat <- tau2*(1-tau2) / f.hat.xi2.hat^2
  v12.hat <- tau1*(1 - tau2) / (f.hat.xi1.hat * f.hat.xi2.hat)
  v.hat <- v1.hat + v2.hat - 2* v12.hat

  asymp.v.hat.lower[s] <- dxi.hat - z_val * sqrt(v.hat) / sqrt_n
  asymp.v.hat.upper[s] <- dxi.hat + z_val * sqrt(v.hat) / sqrt_n

}
mean((asymp.v.hat.lower < dxi) & (asymp.v.hat.upper > dxi))
```

When I varied the sample size n, I got the following table of coverages:

n	20	30	40	50	60	70	80	90	100
coverage	0.880	0.892	0.920	0.930	0.924	0.922	0.934	0.924	0.938

(d) Now consider using the bootstrap to estimate the sampling distributions of the quantities

$$\sqrt{n}[(\hat{\xi}_{\tau_2} - \hat{\xi}_{\tau_1}) - (\xi_{\tau_2} - \xi_{\tau_1})] \quad \text{and} \quad \sqrt{n}\frac{[(\hat{\xi}_{\tau_2} - \hat{\xi}_{\tau_1}) - (\xi_{\tau_2} - \xi_{\tau_1})]}{\sqrt{\hat{\vartheta}}},$$

that is, of the unstudentized and studentized pivots. Do the following:

i. Describe how you construct the bootstrap version of each pivot quantity.

Solution: The bootstrap pivots are

$$\sqrt{n}[(\hat{\xi}^*_{\tau_2} - \hat{\xi}^*_{\tau_1}) - (\hat{\xi}_{\tau_2} - \hat{\xi}_{\tau_1})] \quad \text{and} \quad \sqrt{n}\frac{[(\hat{\xi}^*_{\tau_2} - \hat{\xi}^*_{\tau_1}) - (\hat{\xi}_{\tau_2} - \hat{\xi}_{\tau_1})]}{\sqrt{\hat{\vartheta}^*}},$$

where $\hat{\vartheta}^*$ is just like $\hat{\vartheta}$ but computed on $X^*_1, \ldots, X^*_n$. You may or may not decide to make a new selection of the bandwidth in every bootstrap sample; I chose not to, so for me the bandwidth h was fixed inside the bootstrap loop.

ii. Generate 100 data sets (under the same settings as before) and compute bootstrap-based CIs for the IQR using based on the unstudentized and studentized pivots. Use 500 Monte Carlo draws to approximate the bootstrap distributions.

iii. Report the coverages of these intervals and compare them to that of the non-bootstrap confidence interval from the previous part. Turn in code.

Solution:
My code was

```
alpha <- 1.5
beta <- 6
tau1 <- .25
tau2 <- .75

xi1 <- qgamma(tau1,alpha,scale=beta)
xi2 <- qgamma(tau2,alpha,scale=beta)
dxi <- xi2 - xi1

n <- 20
S <- 100
B <- 500
sqrt_n <- sqrt(n)
z_val <- qnorm(.975)
asymp.v.hat.lower <- asymp.v.hat.upper <- numeric(S)
boot.U.lower <- boot.U.upper <- numeric(S)
boot.S.lower <- boot.S.upper <- numeric(S)
for( s in 1:S ){

  X <- rgamma(n,alpha,scale = beta)

  xi1.hat <- quantile(X,probs=tau1)
  xi2.hat <- quantile(X,probs=tau2)
  dxi.hat <- xi2.hat - xi1.hat

  # choose bandwidth
  h <- density(x = X)$bw
  f.hat.xi1.hat <- mean(dnorm((X - xi1.hat)/h))/h
  f.hat.xi2.hat <- mean(dnorm((X - xi2.hat)/h))/h

  v1.hat <- tau1*(1-tau1) / f.hat.xi1.hat^2
  v2.hat <- tau2*(1-tau2) / f.hat.xi2.hat^2
  v12.hat <- tau1*(1 - tau2) / (f.hat.xi1.hat * f.hat.xi2.hat)
  v.hat <- v1.hat + v2.hat - 2* v12.hat

  asymp.v.hat.lower[s] <- dxi.hat - z_val * sqrt(v.hat) / sqrt_n
  asymp.v.hat.upper[s] <- dxi.hat + z_val * sqrt(v.hat) / sqrt_n

  # now the bootstrap, unstudentized and studentized
  boot.U.star <- numeric(B)
  boot.S.star <- numeric(B)
  for( b in 1:B ){

    X.star <- X[sample(1:n,n,replace=TRUE)]

    # unstudentized, unsmoothed
```

```
    xi1.hat.star <- quantile(X.star,probs=tau1)
    xi2.hat.star <- quantile(X.star,probs=tau2)
    dxi.hat.star <- xi2.hat.star - xi1.hat.star
    boot.U.star[b] <- dxi.hat.star - dxi.hat

    # studentized, unsmoothed
    f.hat.star.xi1.hat.star <- mean(dnorm((X.star - xi1.hat.star)/h))/h
    f.hat.star.xi2.hat.star <- mean(dnorm((X.star - xi2.hat.star)/h))/h

    v1.hat.star <- tau1*(1-tau1) / f.hat.star.xi1.hat.star^2
    v2.hat.star <- tau2*(1-tau2) / f.hat.star.xi2.hat.star^2
    v12.hat.star <- tau1*(1-tau2)/(f.hat.star.xi1.hat.star*f.hat.star.xi2.hat.star)
    v.hat.star <- v1.hat.star + v2.hat.star - 2* v12.hat.star

    boot.S.star[b] <- sqrt_n * ( dxi.hat.star - dxi.hat ) / sqrt(v.hat.star)

  }

  boot.U.star.qtls <- quantile(boot.U.star,probs = c(0.025,0.975))
  boot.S.star.qtls <- quantile(boot.S.star,probs = c(0.025,0.975))

  boot.U.lower[s] <- dxi.hat - boot.U.star.qtls[2]
  boot.U.upper[s] <- dxi.hat - boot.U.star.qtls[1]

  boot.S.lower[s] <- dxi.hat - boot.S.star.qtls[2] * sqrt(v.hat) / sqrt_n
  boot.S.upper[s] <- dxi.hat - boot.S.star.qtls[1] * sqrt(v.hat) / sqrt_n

}

mean((asymp.v.hat.lower < dxi) & (asymp.v.hat.upper > dxi))
mean((boot.U.lower < dxi) & (boot.U.upper > dxi))
mean((boot.S.lower < dxi) & (boot.S.upper > dxi))
```

The non-bootstrap interval had coverage 0.88, the unstudentized bootstrap, 0.79, and the studentized bootstrap, 0.91.

5. (a) Give a set of residuals $\hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_n$, consider generating wild bootstrap residuals $\varepsilon_n^*, \ldots, \varepsilon_n^*$ as follows: First generate $U_1^*, \ldots, U_n^*$ as independent Beta(1/2,3/2) rvs. Then set $\varepsilon_i^* = \hat{\varepsilon}_i \cdot 4(U_i^* - 1/4)$ for $i = 1, \ldots, n$. Show that

$$\begin{aligned} \mathbb{E}_*[\varepsilon_i^*] &= 0 \\ \mathbb{E}_*[(\varepsilon_i^*)^2] &= \hat{\varepsilon}_i^2 \\ \mathbb{E}_*[(\varepsilon_i^*)^3] &= \hat{\varepsilon}_i^3, \end{aligned}$$

which are the moment conditions prescribed for the wild bootstrap in [1].

Solution: For $U \sim \text{Beta}(\alpha, \beta)$ we can work out that

$$\begin{aligned}
\mathbb{E}U &= \alpha(\alpha+\beta)^{-1} \\
\mathbb{E}(U-\mathbb{E}U)^2 &= \alpha\beta[(\alpha+\beta)^2(\alpha+\beta+1)]^{-1} \\
\mathbb{E}(U-\mathbb{E}U)^3 &= 2(\beta-\alpha)\alpha\beta[(\alpha+\beta)(\alpha+\beta+2)(\alpha+\beta+1)]^{-1}.
\end{aligned}$$

Finding the third one is tedious; you could look up the skewness of the beta distribution and then multiply this by the variance raised to the power 3/2. Under $\alpha = 1/4$ and $\beta = 3/4$, we have

$$\begin{aligned}
\mathbb{E}[4(U-1/4)] &= 0 \\
\mathbb{E}[4^2(U-1/4)^2] &= 1 \\
\mathbb{E}[4^3(U-1/4)^3] &= 1,
\end{aligned}$$

which gives the result.

(b) Consider the linear regression model $Y_i = \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$ for $i = 1, \ldots, n$, where $\varepsilon_1, \ldots, \varepsilon_n$ are independent rvs such that with $\mathbb{E}\varepsilon_i = 0$ and $\mathbb{E}\varepsilon_i^2 = \sigma_i^2$ for $i = 1, \ldots, n$. More precisely, let $(X_{11}, X_{12}, Y_1), \ldots, (X_{n1}, X_{n2}, Y_n)$ come from the data generating process in the following code:

```
n <- 30
rho <- .5
X1 <- rnorm(n)
X2 <- rnorm(n,rho*X1,1 - rho^2)
X <- cbind(X1 - mean(X1),X2 - mean(X2))
error <- rgamma(n,.5,scale = 1/(1 + exp(-X[,1]))) - .5 * 1/(1 + exp(-X[,1]))
beta <- c(1,2)
Y <- as.numeric(X %*% beta) + error - mean(error)
```

Letting $\boldsymbol{\beta} = (\beta_1, \beta_2)^T$ and $\hat{\boldsymbol{\beta}}_n$ be the least squares estimator of $\boldsymbol{\beta}$, consider, for a vector $\mathbf{c} \in \mathbb{R}^2$ the pivot

$$\sqrt{n}\mathbf{c}^T(\hat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_n)/\hat{\sigma}_c, \tag{1}$$

where, letting $\mathbf{X}$ be the $n \times 2$ matrix with rows (X_{i1}, X_{i2}), $i = 1, \ldots, n$, and $\hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_n$ be the least-squares residuals,

$$\hat{\sigma}_c^2 = n \cdot \mathbf{c}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \cdot \text{diag}(\hat{\varepsilon}_1^2, \ldots, \hat{\varepsilon}_n^2) \cdot \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{c}.$$

i. Give an expression for the variance of $\sqrt{n}\mathbf{c}^T\hat{\boldsymbol{\beta}}_n$.

Solution: We have

$$\text{Var}(\sqrt{n}\mathbf{c}^T\hat{\boldsymbol{\beta}}_n) = n \cdot \mathbf{c}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \cdot \text{diag}(\sigma_1^2, \ldots, \sigma_n^2) \cdot \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{c}.$$

ii. Give the form of an asymptotic $(1-\alpha) \times 100\%$ CI for $\mathbf{c}^T\boldsymbol{\beta}$, assuming that the pivot in (1) converges in distribution to a standard Normal random variable.

Solution: Provided the pivot is asymptotically standard Normal,

$$\mathbf{c}^T\hat{\boldsymbol{\beta}}_n \pm z_{\alpha/2}\hat{\sigma}_c^2/\sqrt{n}$$

is an asymptotic $(1-\alpha)\times 100\%$ CI for $\mathbf{c}^T\boldsymbol{\beta}$.

iii. Generate 500 data sets as in the R code above, and for a vector $\mathbf{c}$ of your choosing, build 95% confidence intervals for $\mathbf{c}^T\boldsymbol{\beta}$ with each of the 500 data sets. Record the proportion of times the confidence interval captured the true value of $\mathbf{c}^T\boldsymbol{\beta}$.

iv. Do the same thing, but this time build a confidence interval using the wild bootstrap pivot $\sqrt{n}\mathbf{c}^T(\hat{\boldsymbol{\beta}}_n^* - \hat{\boldsymbol{\beta}}_n)/\hat{\sigma}_c^*$ with the bootstrap residuals considered in the first part; use

$$(\hat{\sigma}_c^*)^2 = n\cdot\mathbf{c}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\cdot\mathrm{diag}((\hat{\varepsilon}_1^*)^2,\ldots,(\hat{\varepsilon}_n^*)^2)\cdot\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{c}.$$

On each data set, draw 500 bootstrap samples. Record the coverage of the wild bootstrap confidence interval.

Solution: Here is my code:

```
n <- 30
rho <- .5
beta <- c(1,-1)
cc <- c(1,0)
cc.beta <- as.numeric(cc%*% beta)
z_val <- qnorm(.975)

S <- 100
B <- 500

asymp.lower <- asymp.upper <- numeric(S)
wboot.lower <- wboot.upper <- numeric(S)
for( s in 1:S){

  # generate data
  X1 <- rnorm(n)
  X2 <- rnorm(n,rho*X1,1 - rho^2)
  X <- cbind(X1 - mean(X1),X2 - mean(X2))
  error <- rgamma(n,.5,scale = 1/(1 + exp(-X[,1]))) - .5 * 1/(1 + exp(-X[,1]))
  Y <- as.numeric(X %*% beta) + error - mean(error)

  # compute least squares estimator
  XtX.inv.Xt <- solve(t(X) %*% X) %*% t(X)
  beta.hat <- XtX.inv.Xt %*% Y
  Y.hat <- as.numeric(X %*% beta.hat)
  e.hat <- Y - Y.hat
```

```
  # get confidence interval for contrast, assuming nonconstant variance
  sigma.hat.cc <- sqrt(n*t(cc)%*%XtX.inv.Xt%*%diag(e.hat^2)%*%t(XtX.inv.Xt)%*%cc)
  cc.beta.hat <- t(cc) %*% beta.hat

  asymp.lower[s] <- cc.beta.hat - z_val * sigma.hat.cc / sqrt(n)
  asymp.upper[s] <- cc.beta.hat + z_val * sigma.hat.cc / sqrt(n)


  # get wild bootstrap pivot realizations
  wboot.pivot <- numeric(B)
  for( b in 1:B){

    e.star <- e.hat * 4 * ( rbeta(n, shape1 = 1/2, shape2 = 3/2) - 1/4 )
    Y.star <- Y.hat + e.star
    beta.hat.star <- XtX.inv.Xt %*% Y.star
    Y.hat.star <- as.numeric(X %*% beta.hat.star)
    e.hat.star <- Y.star - Y.hat.star

    sigma.hat.cc.star <- sqrt(n*t(cc)%*%XtX.inv.Xt%*%diag(e.hat.star^2)%*%t(XtX.inv.Xt)%*
    cc.beta.hat.star <- t(cc) %*% beta.hat.star

    wboot.pivot[b] <- sqrt(n)*t(cc)%*%(beta.hat.star-beta.hat)/sigma.hat.cc.star

  }

  # construct wild bootstrap confidence intervals
  wboot.pivot.qtls <- quantile(wboot.pivot, prob = c(0.025, 0.975))
  wboot.lower[s] <- cc.beta.hat - wboot.pivot.qtls[2] * sigma.hat.cc / sqrt(n)
  wboot.upper[s] <- cc.beta.hat - wboot.pivot.qtls[1] * sigma.hat.cc / sqrt(n)


}

asymp.cov <- mean( (asymp.lower < cc.beta) & (asymp.upper > cc.beta) )
asymp.width <- mean(asymp.upper - asymp.lower)

asymp.cov
asymp.width

wboot.cov <- mean( (wboot.lower < cc.beta) & (wboot.upper > cc.beta) )
wboot.width <- mean(wboot.upper - wboot.lower)

wboot.cov
wboot.width
```

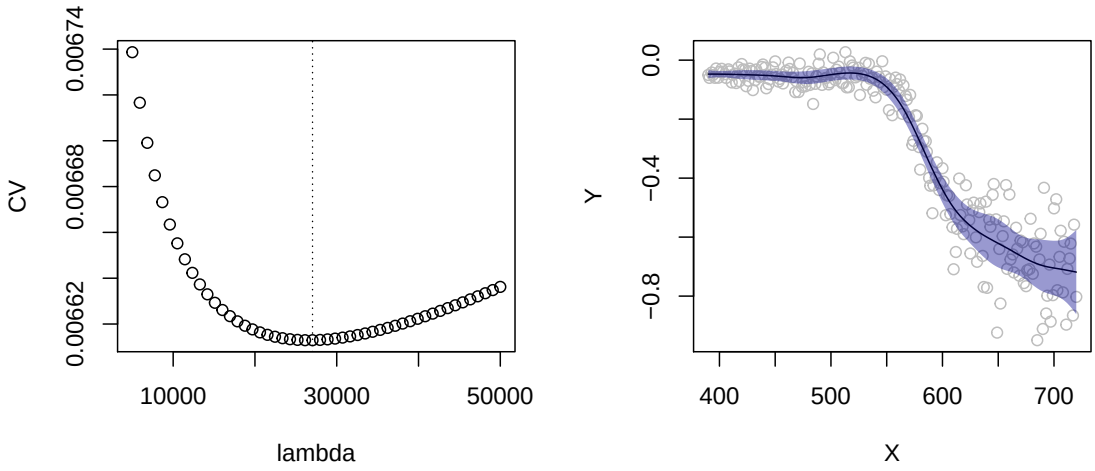
In my simulation the interval based on asympototic Normality achieved coverage 0.866

and the wild bootstrap interval achieved coverage 0.868.

6. Import the `lidar` data set from the R package `SemiPar` and consider a nonparametric regression of `logratio` on `range`.

   (a) Fit a penalized splines estimator using leave-one-out crossvalidation to choose the level of penalization towards smoothness. Use a large number of spline functions in the basis so that the penalized spline estimator will be approximately the same as the smoothing spline estimator.

   (b) Use the tube-formula approach described in Lecture 11 to build a 95% confidence band for the true function, allowing for heterscedastic error term variances. Make plots similar to the ones below, showing the CV output and the fitted function with the confidence band (these are the plots from my analysis; yours should look similar).



Solution: Here is partial code, which makes use of the `pspl` function from here.

```
library(SemiPar)
library(splines)
data(lidar)

X <- lidar$range
Y <- lidar$logratio
n <- length(Y)

# choose lambda by leave-one-out crossvalidation
lambda <- seq(5000,50000, length = 50)
K <- 50
```

```
CV <- numeric(length(lambda))
for( j in 1:length(lambda)){

  pspl.out <- pspl(Y = Y, X = X, K = K, lambda = lambda[j])
  Y.hat <- pspl.out$m.hat(X)
  S <- pspl.out$S

  CV[j] <- mean(   ((Y - Y.hat) / (1 - diag(S)))^2   )

}

lambda.CV <- lambda[which.min(CV)]
pspl.out <- pspl(Y = Y, X = X, K = K, lambda = lambda.CV)
Y.hat <- pspl.out$m.hat(X)
S <- pspl.out$S

# we need this function
find_c0 <- function(x,kappa0,alpha){

  val <-   2*(1 - pnorm(x)) + kappa0 / pi * exp( - x^2 / 2) - alpha
  return(val)

}

# compute kappa0 under heteroscedasticity
e.hat <- as.numeric(Y - Y.hat)
v.X <- sqrt(as.numeric(S^2 %*% e.hat ^2))
T.het.X <- S / v.X * abs(e.hat)
D <- cbind(diag(n-1),0) - cbind(0,diag(n-1))
dT.het.X <- D %*% T.het.X
kappa0.het <- sum(sqrt(apply(dT.het.X^2,1,sum)))
kappa0.het

# find c0 under heteroscedasticity
c0.het <- uniroot(find_c0,kappa0 = kappa0.het,alpha = 0.05,lower = 1,upper = 5)$root

# construct lower and upper limits of confidence band
asymp.het.lower <- Y.hat - c0.het * v.X
asymp.het.upper <- Y.hat + c0.het * v.X
```

References

[1] Enno Mammen. Bootstrap and wild bootstrap for high dimensional linear models. *The Annals of Statistics*, pages 255–285, 1993.