Project 4: the Wages data

Contents

List of Figures

0 Notation

Throughout the project I will be refering to n as the length of the sample: $n = nrow(data)$; $B = 1000$ as the number of bootstrap samples; i as an index for objects in sample: $i = 1, \ldots, n$; b as an index for bootstrap samples: $b = 1, \ldots, B$. By "bootstrap replicate" I mean estimate of statistic computed in one bootstrap sample. By "resampling" I mean drawing objects from the observed sample with replacement.

1 Question 1

This question focuses on 4 variables regarding people: *income* (Y_i), *age* (X_{1i}), *number of children* (X_{2i}), *gender* (X_{3i}) and *martial status* (X_{4i}). Let's consider the model:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4i} + \varepsilon_i$$

Where $\varepsilon_i \sim N(0, \sigma^2)$ and σ is unknown.

1.1

The task is to estimate model and 95% confidence intervals for parameters. We know that using ML (maximum likelihood) or REML (restricted maximum likelihood) methods for estimating parameters, estimators of β_i have asymptotic normal distribution and we can estimate confidence intervals. Table 1.1 presents estimates computed using functions *lm* and *confint*. None of the intervals contain 0 so we can

	lower	estimate	upper	variable
$\hat{\beta_0}$	4243.2717	5537.20	6831.1219	
$\hat{\beta_1}$	293.3845	317.64	341.8947	age
$\hat{\beta_2}$	-1184.9820	-982.59	-780.1930	childs
$\hat{\beta_3}$	11215.7810	11780.41	12345.0443	male
$\hat{\beta_4}$	1425.6156	2251.73	3077.8525	married
	-6474.6473	-5474.41	-4474.1812	never married
	-4336.2567	-2677.56	-1018.8538	separated
	-9138.7566	-7479.31	-5819.8690	widowed

Table 1.1: Table of parameter estimates and its confidence intervals.

infer that every variable is statistically significant.

1.2

The task is to use parametric and non-parametric bootstrap to estimate the standard error and the distribution of $\hat{\beta_1}$ and construct 95% confidence intervals with different methods.

For non-parametric bootstrap we resample $B = 1000$ times tuples $(Y_i, X_{1i}, X_{2i}, X_{3i}, X_{4i})$. For parametric bootstrap we draw $Y_i^{(b)} \sim N(\hat{\beta_0} + \hat{\beta_1} X_{1i} + \hat{\beta_2} X_{2i} + \hat{\beta_3} X_{3i} + \hat{\beta_4} X_{4i}, \hat{\sigma}),\ i = 1, \ldots, n,\ b = 1, \ldots, B$ where estimators $\hat{\beta_i}, \hat{\sigma}$ are computed based on observed sample. For each bootstrap sample we construct model and estimate β_1.

The standard error of estimator ($\hat{\sigma}_1$) is calculated as a square of sample variance of bootstrap replicates.

The improved normal CI are calculated as a difference between observed estimator and its bias and plus/minus its standard error times normal quantiles (bias and standard error are estimated using bootstrap method).

$$\hat{\beta_1}^{obs} - bias \pm \hat{\sigma}_1 \cdot z_{\frac{\alpha}{2}}.$$

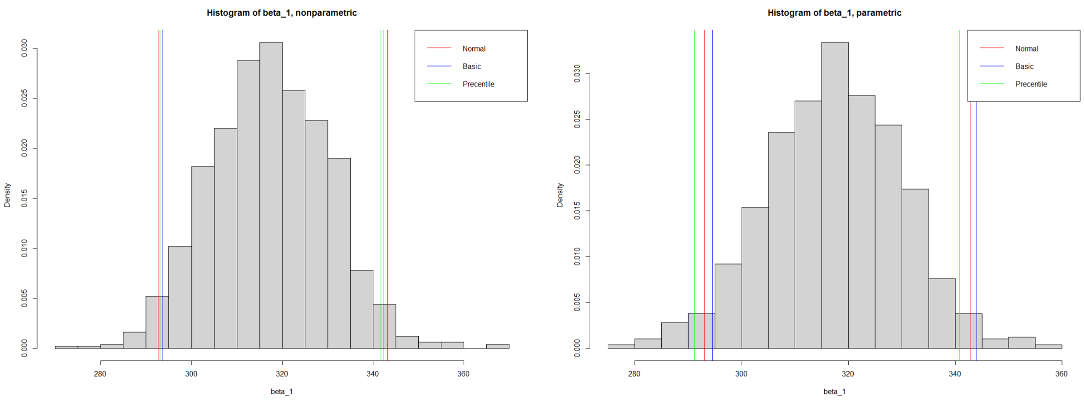
The basic bootstrap CI are calculated as an 2 times observed estimator plus/minus quantiles of the estimator obtained through the boostrap replicates. The percentile bootstrap CI are calculated as a quantiles from the bootstrap replicates.

$$\left[2\hat{\beta_1}^{obs} - \hat{\beta_1}^*_{(B+1)\alpha}, 2\hat{\beta_1}^{obs} - \hat{\beta_1}^*_{(B+1)(1-\alpha)}\right], \quad \left[\hat{\beta_1}^*_{(B+1)\alpha}, \hat{\beta_1}^*_{(B+1)(1-\alpha)}\right]$$

All of the CI are presented in the table 1.2. The distribution of $\hat{\beta_1}$ is presented on a figure 1.1. The CIs are all close to each other. In case of non-parametric bootstrap the improved normal CIs are the widest. The basic and percentile CIs are shifted with respect to each other.

	non-parametric		parametric	
CI	lower	upper	lower	upper
improved normal	292.6807	343.1661	293.0706	342.9147
basic bootstrap	293.5223	342.2225	294.4930	344.0445
percentile	293.0566	341.7569	291.2346	340.7862

Table 1.2: Table of confidence intervals of β_1.



(a) Non-parametric bootstrap, $\hat{\sigma}_1 = 12.88$. (b) Parametric bootstrap, $\hat{\sigma}_1 = 12.72$.

Figure 1.1: Distribution of estimator $\hat{\beta}_1$.

1.3

The task is to use parametric, non-parametric and semi-parametric bootstrap to test the null hypothesis: $\beta_2 = \beta_3 = 0$ using likelihood ratio test. The test statistic has form:

$$T = -2\left(l^r - l^s\right) \underset{H_0}{\rightarrow} \chi^2_k.$$

Where l^r is log-likelihood of reduced model (H_0), l^s is log-likelihood of saturated model (H_1) and $k = 2$ is difference in degrees of freedom between models.

The parametric and non-parametric bootstraps we resample $B = 1000$ times, the same way as in previous task but they need to hold the distribution of H_0. For parametric bootstrap we draw $Y_i^{(b)} \sim N(\hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_4 \hat{X}_{4i}, \hat{\sigma}),\ i = 1, \ldots, n,\ b = 1, \ldots, B$ where estimators $\hat{\beta}_i, \hat{\sigma}$ are computed based on observed sample.

For semi-parametric bootstrap we obtain residuals $\hat{e}_i = Y_i - (\hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_4 \hat{X}_{4i})$ where estimators $\hat{\beta}_i$ are computed based on observed sample. Then we resample $\hat{e}_i$ and obtain new Y as: $Y_i^{(b)} = \hat{e}_i{}^* + (\hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_4 X_{4i}),\ i = 1, \ldots, n,\ b = 1, \ldots, B$.

For non-parametric bootstrap we fix X_{2i}, X_{3i} and resample tuples (Y_i, X_{1i}, X_{4i}).

Now with resampled data we construct reduced and saturated models, so that we can compute their log-likelihoods and then test statistic. Computing *Monte Carlo p-value*:

$$P = \frac{\#\left\{|\hat{T^b}| \geq |\hat{T^{obs}}|\right\} + 1}{B + 1} = 0.000999$$

Every method gave the same result, rejecting H_0, which means, that *childs* and *gender* have influence on *income*. The first task in this question suggested likewise, confidence intervals of those variables were far from zero.

1.4

The task is to test the hypothesis: $\beta_2 = \beta_3 = 0$ using permutation test. To do that, we are following the same steps as in previous task with non-parametric bootstrap, but this time resampling is done without replacement. *Monte Carlo p-value* equals 0.000999 so we reject hypothesis that two parameters are equal 0.

2 Question 2

This question focuses on the same variables and model as in Question 1.

2.1

The task is to predict income of a new subject and construct 95% confidence intervals for this prediction using parametric and non-parametric bootstrap. For non-parametric bootstrap we simply resample ($B =$ 1000 times) tuples $(Y_i, X_{1i}, X_{2i}, X_{3i}, X_{4i})$ and for parametric bootstrap we draw $Y_i^{(b)} \sim N(\hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \hat{\beta}_3 X_{3i} + \hat{X}_{4i}, \hat{\sigma}),\ i = 1, \ldots, n,\ b = 1, \ldots, B$ where estimators $\hat{\beta}_i, \hat{\sigma}$ are computed based on observed sample. Then from bootstrap samples we construct a new model and get the prediction.

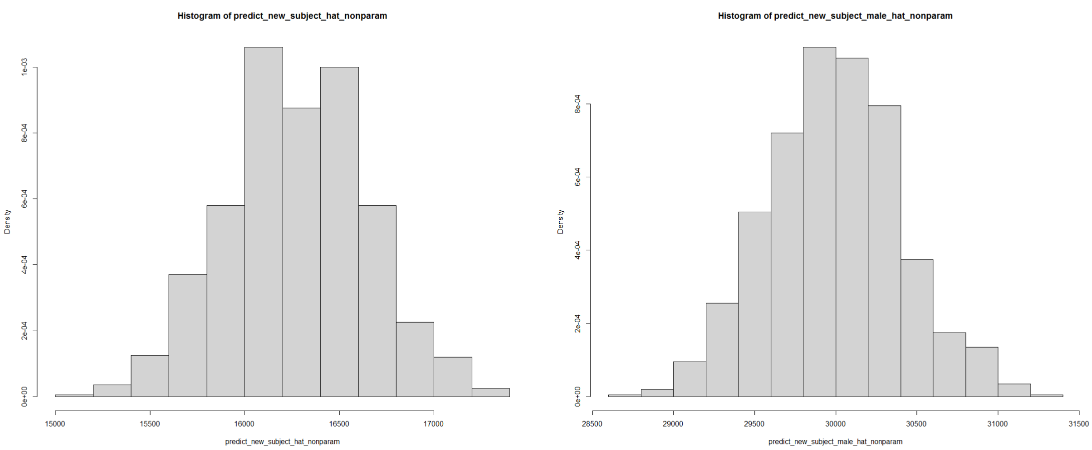
The estimator of this prediction is a sample mean from bootstrap replicates and its confidence intervals are computed as quantiles (bootstrap percentile CI). The results are presented in table 2.1.

	lower	estimate	upper
non-parametric	15553.44	16274.54	17004.45
parametric	15519.07	16294.81	17048.41

Table 2.1: Table of confidence intervals for prediction.

2.2

The task is to predict expected income for another subject and compare it to the previous one. It is done exactly like in previous task. The bootstrap precentile CI: [29218.7130840.01]. Distributions of those two subjects are presented on a figure 2.1. For male, expected income is significantly higher than for female, also the CI are narrower.



(a) Female, age 40, 2 children, divorced

(b) Male, age 46, 2 children, divorced

Figure 2.1: Distributions of expected income for 2 subjects.

2.3

The task is to estimate standard error for predicted income of the two individuals using semi-parametric bootstrap. For semi-parametric bootstrap we obtain residuals $\hat{e}_i = Y_i - (\hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \hat{\beta}_3 X_{3i} + \hat{\beta}_4 X_{4i)}$ where estimators $\hat{\beta}_i$ are computed based on observed sample. Then we resample $\hat{e}_i$ and obtain new Y as: $Y_i^{(b)} = \hat{e}_i{}^* + (\hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \hat{\beta}_3 X_{3i} + \hat{\beta}_4 X_{4i}),\ i = 1, \ldots, n,\ b = 1, \ldots, B$. Estimate of standard errors are calculated as a square root of sample variance of bootstrap replicates and are presented in table 2.2.

	Female	Male
$\hat{s.e.}$	387.4626	406.8205

Table 2.2: Table of standard errors of predicted income.

3 Question 3

This question focuses on *gender* (Y_i) and *childs3* – indicator variable which takes the value of 1 if the subject has less than 3 children and zero otherwise (X_i). Let π_F and π_M be the proportion of female and male with less than 3 children respectively. Data used in this question omitted *NA's*.

3.1

The task is to estimate π_F and π_M and construct 95% confidence intervals for difference $\pi_F = \pi_M$ using classical methods. Estimator of π_F is a number of females with less than 3 children divided by the number of females and $\hat{\pi_F} = 0.73$. Similarly $\hat{\pi_M} = 0.74$. For the hypothesis $H_0 : \ \pi_F = \pi_M$ we will be using statistic:

$$t = \frac{\hat{\pi_F} - \hat{\pi_M}}{\sqrt{\hat{\pi}(1-\hat{\pi})(\frac{1}{n} + \frac{1}{m})}} \overset{H_0}{\sim} N(0,1).$$

Where n, m are numbers of females and males respectively and $\hat{\pi} = (n\hat{\pi_F} + m\hat{\pi_M})/(n+m)$. The test statistic under null is from standard normal distribution, so we can construct theoretical CI:

$$\hat{\pi_F} - \hat{\pi_M} \pm z_{1-\frac{\alpha}{2}} \cdot \sqrt{\hat{\pi}(1-\hat{\pi})\left(\frac{1}{n} + \frac{1}{m}\right)}.$$

Where $z_{1-\frac{\alpha}{2}}$ is standard normal quantile. With $\hat{\pi} = 0.7364798$ we can compute confidence intervals: $[-0.0162, 0.0017]$. They contain 0, so we accept null hypothesis.

3.2

The task is to construct 95% confidence intervals for difference $\pi_F - \pi_M$ using parametric bootstrap. For parametric bootstrap we assume that $f_i \sim b(1, \hat{\pi_F}),\ i = 1, \ldots, n$ and $m_i \sim b(1, \hat{\pi_M}),\ i = 1, \ldots, m$, where those variables are indicators which take value 1 when female or male has less than 3 children respectively. We draw $B = 1000$ samples from those distributions and estimate the difference.

The percentile bootstrap CI are calculated as a quantiles from the bootstrap replicates and are equal: $[-0.0164, 0.0015]$. These are narrower than the theoretical ones.

3.3

The task is to test the null hypothesis $H_0 : \ \pi_F = \pi_M$ using non-parametric bootstrap. To do that, we create artificially vectors f, m that are n and m long and have the same number of ones as the number of females and males who have less than 3 children respectively (the rest filled with zeros). For non-parametric bootstrap we resample joint vector (f, m) (so that H_0 holds).

Given vector (f, m) we compute statistics $t^{(b)}$. *Monte Carlo p-value* equals 0.1168831 so we can accept (depending on significance level) hypothesis that proportions of male and female having less than 3 children is equal.

3.4

The task is to compare the distribution of the test statistics in previous task with the asymptotic distribution of this statistic (standard normal under null). Given bootstrap replicates $t^{(b)}$ (computed in previous task) we can draw histogram and compare it to the standard normal density. Histogram is presented on a figure 3.1. From the histogram, we can see that it matches theoretical density.

4 Question 4

This question focuses on variable *occupational prestige score* (Y_i).

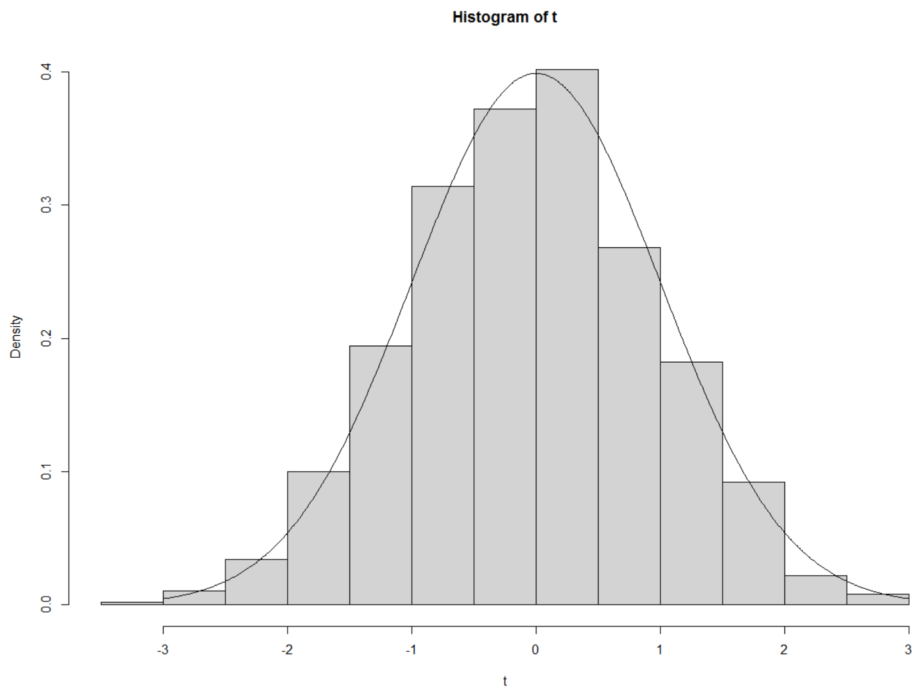


Figure 3.1: Histogram of test statistic t (under null).

4.1

The task is to estimate the mean *occupational prestige score* and to construct 95% confidence intervals for the mean and the variance using classical methods. The estimator of the mean is a sample mean. Assuming normality (and not knowing parameters of distribution) of variable we can construct CIs:

$$\mu: \ \bar{X} \pm t_{n-1} \cdot \frac{S^2}{\sqrt{n}}, \quad \sigma^2: \left[\frac{(n-1)S^2}{\chi^2_{\frac{\alpha}{2},n-1}}, \frac{(n-1)S^2}{\chi^2_{\frac{1-\alpha}{2},n-1}} \right].$$

Where S^2 is a sample variance, t_{n-1} is $\alpha/2$ quantile from Student's t-distribution with $n-1$ degrees of freedom $\chi^2_{\alpha/2,n-1}$ is $\alpha/2$ quantile from χ^2 distribution with $n-1$ degrees of freedom. Estimates and CIs are presented in table 4.1.

	lower	estimate	upper
μ	43.69080	43.82383	43.95685
σ^2	168.8478	171.2835	173.7725

Table 4.1: Table of estimates and confidence intervals of parameters.

4.2

The task is to construct 95% confidence intervals for the mean and the variance of *occupational prestige score* using parametric bootstrap. For this method we assume that variable is from $N(\hat{\mu}, \hat{\sigma^2})$ where $\hat{\mu}, \hat{\sigma^2}$ are sample mean and variance from observed sample. Then we draw $B = 1000$ bootstrap samples from this distribution and compute sample mean and variance.

The percentile bootstrap CIs are calculated as a quantiles from the bootstrap replicates and are presented in table 4.2. These are narrower than the theoretical ones.

	lower	upper
μ	43.70096	43.95477
σ^2	168.9597	173.6888

Table 4.2: Table of confidence intervals of parameters estimated using parametric bootstrap.

4.3

The task is to test the hypothesis $H_0: \ \mu_Y = 43.14, \quad H_1: \ \mu_Y < 43.14.$ using classical methods. To do that we can use t-statistic:

$$t = \frac{\bar{X} - 43.14}{\sqrt{S^2}} \overset{H_0}{\sim} t_{n-1}.$$

Where S^2 is sample variance, t_{n-1} Student's t distribution with $n-1$ degrees of freedom. The p-value for observed data (with adequate side, given the alternative) is equal 1, so we accept that $\mu_Y = 43.14$

4.4

The task is to to test the hypothesis $H_0: \ \mu_Y = 43.14, \quad H_1: \ \mu_Y < 43.14.$ using non-parametric bootstrap. For non-parametric bootstrap we need to transform the data, so that it holds for null hypothesis. We will transform $\tilde{Y_i} = Y_i - \bar{Y} + 43.14$. Now we can resample $\tilde{Y_i}$ and make $B = 1000$ bootstrap samples, which will be used to compute t-statistics ($t^b, \ b = 1, \dots B$). Calculating *Monte Carlo p-value*:

$$P = \frac{\#\left\{\hat{t^b} \leq \hat{t^{obs}}\right\} + 1}{B + 1} = 1$$

Again, p-value is equal 1, so we accept that $\mu_Y = 43.15$

Appendices

Appendix 1 Bootstrap algorithm

```
##########################################################################################
###########################BOOTSTRAP######################################################

#BOOTSTRAP ALGORITHM FOR A VECTOR
resample_vector_nonparam = function(X, n=length(X)) sample(X, size = n,
    replace = TRUE)

resample_vector_param = function(X, rdist, n=length(X)) rdist(n)

bootstrap_vector = function(B=1000, X, theta_est, param=FALSE, rdist){
  if(param)
    X_boot = sapply(1:B, function(n)resample_vector_param(X, rdist, length(
        X)))
  else
    X_boot = sapply(1:B, function(n)resample_vector_nonparam(X, length(X)))

  theta_hat = apply(X_boot, 2, theta_est)
  return(theta_hat)
}

#BOOTSTRAP ALGORITHM FOR A DATAFRAME
resample_dataframe_nonparam = function(data, cols=1:ncol(data), n=nrow(data
    ), replacement=TRUE){
  id_boot = sample(1:n, size = n, replace = replacement)
  data[,cols] = data[id_boot, cols]
  return(data)
}

resample_dataframe_param = function(data, rdist_list, cols=1, n=nrow(data))
    {
  for (col in cols)
    data[,col] = rdist_list[[col]](n)

  return(data)
}

bootstrap_dataframe = function(B=1000, data, theta_est, cols=1:ncol(data),
    param=FALSE, rdist_list, replacement=TRUE){
  if(param)
    data_boot = lapply(1:B, function(n)resample_dataframe_param(data, rdist
        _list, cols, nrow(data)))
  else
    data_boot = lapply(1:B, function(n)resample_dataframe_nonparam(data,
        cols, nrow(data), replacement))

  theta_hat = sapply(data_boot, theta_est)
  return(theta_hat)
}

#MONTE CARLO P-VALUE
p_value_boot = function(theta_hat, theta_obs)
```

Appendix 2 Confidence intervals

```
##################################################################################
################BOOTSTRAP CONFIDENCE INTERVALS###############################
improved_normal_CI = function(theta_boot, theta_obs, alpha=.05){
  bias = mean(theta_boot) - theta_obs
  se = sd(theta_boot)
  return( theta_obs - bias + se * qnorm(c(alpha/2, 1-alpha/2)) )
}

basic_bootstrap_CI = function(theta_boot, theta_obs, alpha=.05){
  return( 2*theta_obs - quantile(theta_boot, probs = c(1-alpha/2, alpha/2))
       )
}

percentile_CI = function(theta_boot, alpha=.05){
  return( quantile(theta_boot, probs = c(alpha/2, 1-alpha/2)) )
}
```

Appendix 3 Preparing data

```
##################################################################################
#########################PROJECT 1#########################################
#Preparing data
set.seed(297759)
library(stevedata)
data(gss_wages)
names(gss_wages)
gss_wages = na.omit(gss_wages)
attach(na.omit(gss_wages))
```

Appendix 4 Question 1

```
#Question 1
#Q1.1
fit = lm(realrinc~age+childs+gender+maritalcat)
confint(fit)
#Q1.2
#nonparametric
beta_est = function(data){
  lm(data$realrinc~data$age+data$childs+data$gender+data$maritalcat)$
      coefficients
}
beta_hat_nonparam = bootstrap_dataframe(B=1000, data = data.frame(realrinc,
     age, childs, gender, maritalcat),
                                        theta_est = beta_est)

save(beta_hat_nonparam, file="CIM_project4_1_2_beta_hat_nonparam.RData")

#Distribution
beta_1_hat = beta_hat_nonparam[2,]
sd(beta_1_hat)
hist(beta_1_hat, freq = FALSE, main = "Histogram of beta_1, nonparametric",
     xlab = "beta_1", breaks=20)
#CI's
beta_1_obs = fit$coefficients[2]
se_beta_1_obs = sqrt(vcov(fit)[2,2])
improved_normal_CI(beta_1_hat, beta_1_obs) #red
```

```
basic_bootstrap_CI(beta_1_hat, beta_1_obs) #blue
percentile_CI(beta_1_hat) #green
studentized_CI(beta_1_hat, beta_1_obs, se_beta_1_obs) #purple == green?
legend("topright", legend=c("Normal", "Basic", "Precentile"), col = c("red"
    , "blue", "green"), lty=1)

#Parametric
realrinc_dist = function(n) rnorm(n, predict(fit), summary(fit)$sigma)
beta_hat_param = bootstrap_dataframe(B=1000, data = data.frame(realrinc,
    age, childs, gender, maritalcat),
                                     theta_est = beta_est, param = TRUE,
                                         cols = 1, rdist_list = list(
                                         realrinc_dist))
save(beta_hat_param, file="CIM_project4_1_2_beta_hat_param.RData")
#Distribution
beta_1_hat = beta_hat_param[2,]
sd(beta_1_hat)
hist(beta_1_hat, freq = FALSE, main = "Histogram␣of␣beta_1,␣parametric",
    xlab = "beta_1", breaks=20)
#CI's
improved_normal_CI(beta_1_hat, beta_1_obs) #red
basic_bootstrap_CI(beta_1_hat, beta_1_obs) #blue
percentile_CI(beta_1_hat) #green
studentized_CI(beta_1_hat, beta_1_obs, se_beta_1_obs) #purple == green?
legend("topright", legend=c("Normal", "Basic", "Precentile"), col = c("red"
    , "blue", "green"), lty=1)

#Q1.3
loglik_test = function(data){
  fit_saturated = lm(data$realrinc~data$age+data$childs+data$gender+data$
      maritalcat)
  fit_reduced = lm(data$realrinc~data$age+data$maritalcat)
  return( -2 * as.numeric(logLik(fit_reduced) - logLik(fit_saturated)) )
}

#Nonparametric
loglik_hat_nonparam = bootstrap_dataframe(B=1000, data = data.frame(
    realrinc, age, childs, gender, maritalcat),
                                          theta_est = loglik_test, cols = c
                                              (1,2,5))
save(loglik_hat_nonparam, file="CIM_project4_1_3_loglik_hat_nonparam.RData"
    )
#Monte Carlo p-value
fit_reduced = lm(realrinc~age+maritalcat)
loglik_obs = -2 * as.numeric(logLik(fit_reduced) - logLik(fit))
p_value_boot(loglik_hat_nonparam, loglik_obs) #reject H_0

#Parametric
#to obtain beta_0 + beta_1 X_1i + beta_4 X_4i, without creating matrix X
    with many dummy variables
means_reduced = predict(fit, newdata = data.frame(realrinc, age, childs=0,
    gender="Female", maritalcat))
realrinc_reduced_dist = function(n) rnorm(n, means_reduced, summary(fit)$
    sigma)
loglik_hat_param = bootstrap_dataframe(B=1000, data = data.frame(realrinc,
    age, childs, gender, maritalcat),
                                       theta_est = loglik_test, param =
                                           TRUE, cols = 1, rdist_list = list
```

```
                                         (realrinc_reduced_dist))
save(loglik_hat_nonparam, file="CIM_project4_1_3_loglik_hat_param.RData")

#Monte Carlo p-value
p_value_boot(loglik_hat_param, loglik_obs) #reject H_0

#Semi-parametric
loglik_hat_semiparam_est = function(e){
  y = e + predict(fit, newdata = data.frame(realrinc, age, childs=0, gender
      ="Female", maritalcat))
  return( loglik_test(data.frame(realrinc=y, age, childs, gender,
      maritalcat)) )
}
loglik_hat_semiparam = bootstrap_vector(B=1000, X = fit$residuals, theta_
    est = loglik_hat_semiparam_est)
save(loglik_hat_semiparam, file="CIM_project4_1_3_loglik_hat_semiparam.
    RData")

#Monte Carlo p-value
p_value_boot(loglik_hat_semiparam, loglik_obs) #reject H_0

#Q1.4
loglik_hat_nonparam_perm = bootstrap_dataframe(B=1000, data = data.frame(
    realrinc, age, childs, gender, maritalcat),
                                         theta_est = loglik_test, cols = c
                                           (1,2,5), replacement = FALSE)
save(loglik_hat_nonparam_perm, file="CIM_project4_1_4_loglik_hat_nonparam_
    perm.RData")

#Monte Carlo p-value
p_value_boot(loglik_hat_nonparam_perm, loglik_obs) #reject H_0
```

Appendix 5 Question 2

```
#Question 2
#Q2.1
fit = lm(realrinc~age+childs+gender+maritalcat)
new_subject = data.frame(age=40, childs=2, gender="Female", maritalcat="
    Divorced")
predict(fit, newdata = new_subject, interval = "confidence")

predict_new_subject_est = function(data){
  new_fit = lm(realrinc~age+childs+gender+maritalcat, data = data)
  return(predict(new_fit, newdata = new_subject))
}

#Non-parametric
predict_new_subject_hat_nonparam = bootstrap_dataframe(B=1000, data = data.
    frame(realrinc, age, childs, gender, maritalcat),
                                             theta_est = predict_new_
                                               subject_est)
save(predict_new_subject_hat_nonparam, file="CIM_project4_2_1_predict_new_
    subject_hat_nonparam.RData")
percentile_CI(predict_new_subject_hat_nonparam)

#Parametric
predict_new_subject_hat_param = bootstrap_dataframe(B=1000, data = data.
    frame(realrinc, age, childs, gender, maritalcat),
```

```
                                        theta_est = predict_
                                          new_subject_est ,
                                        param = TRUE, rdist_
                                          list = list(realrinc
                                          _dist), cols = 1)
save(predict_new_subject_hat_param, file="CIM_project4_2_1_predict_new_
    subject_hat_param.RData")
percentile_CI(predict_new_subject_hat_param)

#Q2.2
predict_new_subject_male_est = function(data){
  new_fit = lm(realrinc~age+childs+gender+maritalcat, data = data)
  return(predict(new_fit, newdata = data.frame(age=46, childs=2, gender="
      Male", maritalcat="Divorced")))
}
#Non-parametric
predict_new_subject_male_hat_nonparam = bootstrap_dataframe(B=1000, data =
    data.frame(realrinc, age, childs, gender, maritalcat),
                                        theta_est = predict_
                                          new_subject_male_
                                          est)
save(predict_new_subject_male_hat_nonparam, file="CIM_project4_2_2_predict_
    new_subject_male_hat_nonparam.RData")
percentile_CI(predict_new_subject_male_hat_nonparam)

hist(predict_new_subject_hat_nonparam, freq = FALSE)
hist(predict_new_subject_male_hat_nonparam, freq = FALSE)

#Q2.3
predict_female_semiparam_est = function(e){
  y = e + predict(fit)
  return( predict_new_subject_est(data.frame(realrinc=y, age, childs,
      gender, maritalcat)) )
}
predict_male_semiparam_est = function(e){
  y = e + predict(fit)
  return( predict_new_subject_male_est(data.frame(realrinc=y, age, childs,
      gender, maritalcat)) )
}

predict_new_subject_hat_semiparam = bootstrap_vector(B=1000, X = fit$
    residuals, theta_est = predict_female_semiparam_est)
predict_new_subject_male_hat_semiparam = bootstrap_vector(B=1000, X = fit$
    residuals, theta_est = predict_male_semiparam_est)

save(predict_new_subject_hat_semiparam , file="CIM_project4_2_3_predict_new
    _subject_hat_semiparam.RData")
save(predict_new_subject_male_hat_semiparam , file="CIM_project4_2_3_
    predict_new_subject_male_hat_semiparam.RData")

sd(predict_new_subject_hat_semiparam)
sd(predict_new_subject_male_hat_semiparam)
```

Appendix 6 Question 3

```
#Question 3
table(childs, gender) #with na.omit less than in .pdf
gss_wages$childs3 = (childs < 3)
```

```
attach(gss_wages)

#Q3.1
cont_tab = table(gender, childs3)
n = sum(cont_tab[1,])
m = sum(cont_tab[2,])
pi_obs = prop.table(cont_tab, margin = 1)[,2]
pi_tot_obs = (13534 + 13852) / sum(cont_tab)
pi_obs[1] - pi_obs[2] + qnorm(c(0.025, 0.975)) * sqrt( pi_tot_obs * (1-pi_
    tot_obs) * (1/n + 1/m ) )

#Q3.2
#Parametric
t = NULL
for (i in 1:1000) {
  pi_hat1 = sum(rbinom(n, size = 1, prob = pi_obs[1])) / n
  pi_hat2 = sum(rbinom(m, size = 1, prob = pi_obs[2])) / m
  pi_tot = (n*pi_hat1 + m*pi_hat2) / (n+m)
  t[i] = (pi_hat1 - pi_hat2) #/ sqrt( pi_tot * (1-pi_tot) * (1/n + 1/m ) )
}
percentile_CI(t)

#Q3.3
#Non-parametric under null
z = c(rep(1, 13534), rep(0, n-13534), rep(1, 13852), rep(0, m-13852))
t = NULL
for (i in 1:1000) {
  z_boot = sample(z, size = n+m, replace = TRUE)
  pi_hat1 = sum(z_boot[1:n]) / n
  pi_hat2 = sum(z_boot[(n+1):(n+m)]) / m
  pi_tot = (n*pi_hat1 + m*pi_hat2) / (n+m)
  t[i] = (pi_hat1 - pi_hat2) / sqrt( pi_tot * (1-pi_tot) * (1/n + 1/m ) )
}
p_value_boot(t, (pi_obs[1] - pi_obs[2]) / sqrt( pi_tot_obs * (1-pi_tot_obs)
    * (1/n + 1/m )) )
#high p-value, accept H_0

#Q3.4
hist(t, freq = FALSE)
x = seq(-3,3,by=0.01)
lines(x, dnorm(x))
```

Appendix 7 Question 4

```
#Question 4
#Q4.1
t.test(prestg10)
n = length(prestg10)
(n-1)*var(prestg10) / qchisq(0.975, n-1)
var(prestg10)
(n-1)*var(prestg10) / qchisq(0.025, n-1)

#Q4.2
prestg10_dist = function(n) rnorm(n, mean(prestg10), sd(prestg10))
mean_and_var_est = function(X) return( c(mean(X), var(X)) )
#Parametric
mean_and_var_hat_param = bootstrap_vector(B=1000, prestg10, theta_est =
    mean_and_var_est, param = TRUE, rdist = prestg10_dist)
```

```
save(mean_and_var_hat_param , file="CIM_project4_4_2_mean_and_var_hat_param
    .RData")
percentile_CI(mean_and_var_hat_param[1,])
percentile_CI(mean_and_var_hat_param[2,])

#Q4.3
mu_0 = 43.14
t.test(prestg10, alternative = "less", mu = mu_0)
t_obs = t.test(prestg10, alternative = "less", mu = mu_0)$statistic

#Q4.4
t_est = function(X) t.test(X, alternative = "less", mu = mu_0)$statistic
#Non-parametric
t_hat_nonparam = bootstrap_vector(B=1000, prestg10 - mean(prestg10) + mu_0,
     theta_est = t_est)
save(t_hat_nonparam , file="CIM_project4_4_4_t_hat_nonparam.RData")
(1 + sum(t_hat_nonparam < t_obs)) / (1001)
```