Assignment 2

NAME: Nicholas Tolley
DUE DATE: February 28th, 6pm

Problem 1 (100 pts)

In the earnings dataset you can find salary (*earn*) and some socio-demographic characteristics of each subject, including variables such as *height*, *weight*, gender (*male*), *ethnicity*, *education*, mother's (*mother_education*) and father's education (*father_education*), *walk* (e.g. walking time), *exercise*, if they smoke or not (*smokenow*), *tense*, *angry* and *age*.

The dataset can be found in Canvas in the Data folder (file name: earnings.csv):

(a) (10 points) Subset the data and consider only the variables: *education*, *mother_education*, *father_education*, *walk*, *exercise*, *tense*, *angry*, *weight*, *height*. Check the correlation by performing a figure similar to Figure 1 below (make sure not to use the default colours but rather choose your own). Take special care to the labels and legend. What can you say about the results? What would you expect from a linear regression model (hint: there are some variables to be excluded/included in the model)? Perform a test statistic for the correlation between earn and education, write the hypothesis test and the results you will obtain.

```
library(ggplot2)
library(boot)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## v purrr   1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(tidyr)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'lattice'
##
## The following object is masked from 'package:boot':
##
##     melanoma
##
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
df <- read.csv('earnings.csv')
```

The code below stores a subset of the dataset with the columns indicated above. Since we are calculating the correlation between columns, rows with missing values in any column are removed.

```
subset_cols <- c('education', 'mother_education', 'father_education',
                 'walk', 'exercise', 'tense', 'angry', 'weight', 'height')
label_names <- c('Education', 'Mot._Education', 'Fat._Education', 'Walk', 'Exercise', 'Tense',
                 'Angry', 'Weight', 'Height')
df_subset <- subset(df, select=subset_cols)
df_subset <- drop_na(df_subset)
colnames(df_subset) <- label_names
```

Next we can calculate the correlation between the columns contained in the subset, and visualize the result as a heatmap.

```
corr_matrix <- cor(df_subset)
ggcorr(df_subset, label=TRUE,  low="steelblue", mid="white", high="darkred")
```

As we can see there are several variables that exhibit a near perfect correlation with one another. The most highly correlated columns include:

education <-> mother_education

education <-> father_education

father_education <-> mother_education

tense <-> angry

If we were to build a linear regression model, we would need to remove 2 of the education variables, and either the tense or angry variable. This is because the dataset exhibits what is known as multicollinearity, in other words there is redundant information in the columns. If we were to try and create a linear model on the full dataset, there would not be a unique combination of regression ("beta") coefficients that minimize the residual error. For example, the same coefficient could be assigned tot he angry or tense columns.

The code below calculates the correlation coefficient, and associated p-value, between the earn and education columns. The p-value refers to the probability of the null hypothesis that these two variables are uncorrelated (correlation=0).

```
cor.test(df$earn, df$education)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$earn and df$education
## t = 13.748, df = 1812, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  0.2650564 0.3484265
## sample estimates:
##       cor
## 0.3073311
```

We can see that despite the correlation being relatively low (cor=0.3), the result is highly significant with a p-value $<$ 2.2e-16. We can therefore reject the null hypothesis that the "earn" and "education" variables are uncorrelated with one another.

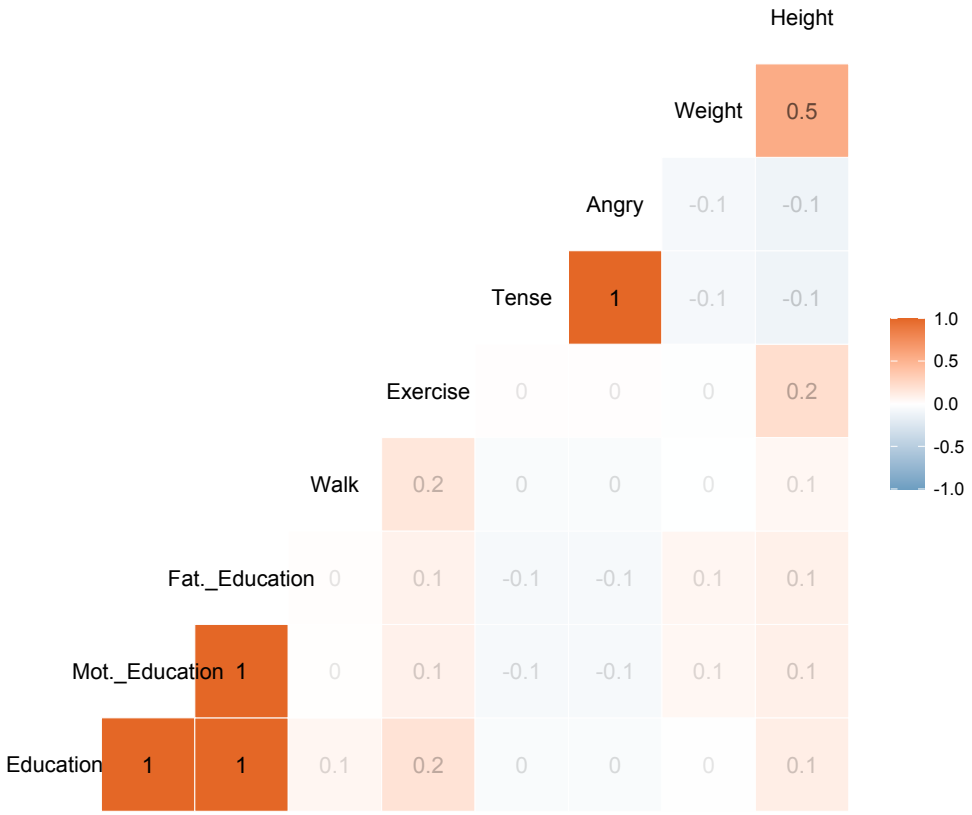


Figure 1: Correlation

(b) (10 points) Perform a linear regression model using the variable *earn* as the dependent variable and years of eductaion *education* as the independent variable. What can you say about this covariate? Is it significant? Write down the hypothesis test. Plot the linear regression you have obtained in ggplot by using a subset of the data. This subset is obtained by restricting the variable *earn* to be less than 2e+05 (similar to Figure 2 below)

The code below creates a linear model that predicts "earn" by the education covariate. From the linear model there are three potential hypothesis tests with the following null hypotheses:

- The intercept of the linear model is zero
- The beta coefficient for education of the linear model is zero

- The model has the same explanatory power (residual error) as constant line set to the mean of the data

```
earn_fit <- lm(df$earn ~ df$education)
summary(earn_fit)
```

```
##
## Call:
## lm(formula = df$earn ~ df$education)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -34051 -12373  -3212   7207 382207
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -14724.1     2657.0  -5.542 3.43e-08 ***
## df$education   2709.7      197.1  13.748  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21460 on 1812 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.09445,    Adjusted R-squared:  0.09395
## F-statistic:   189 on 1 and 1812 DF,  p-value: < 2.2e-16
```

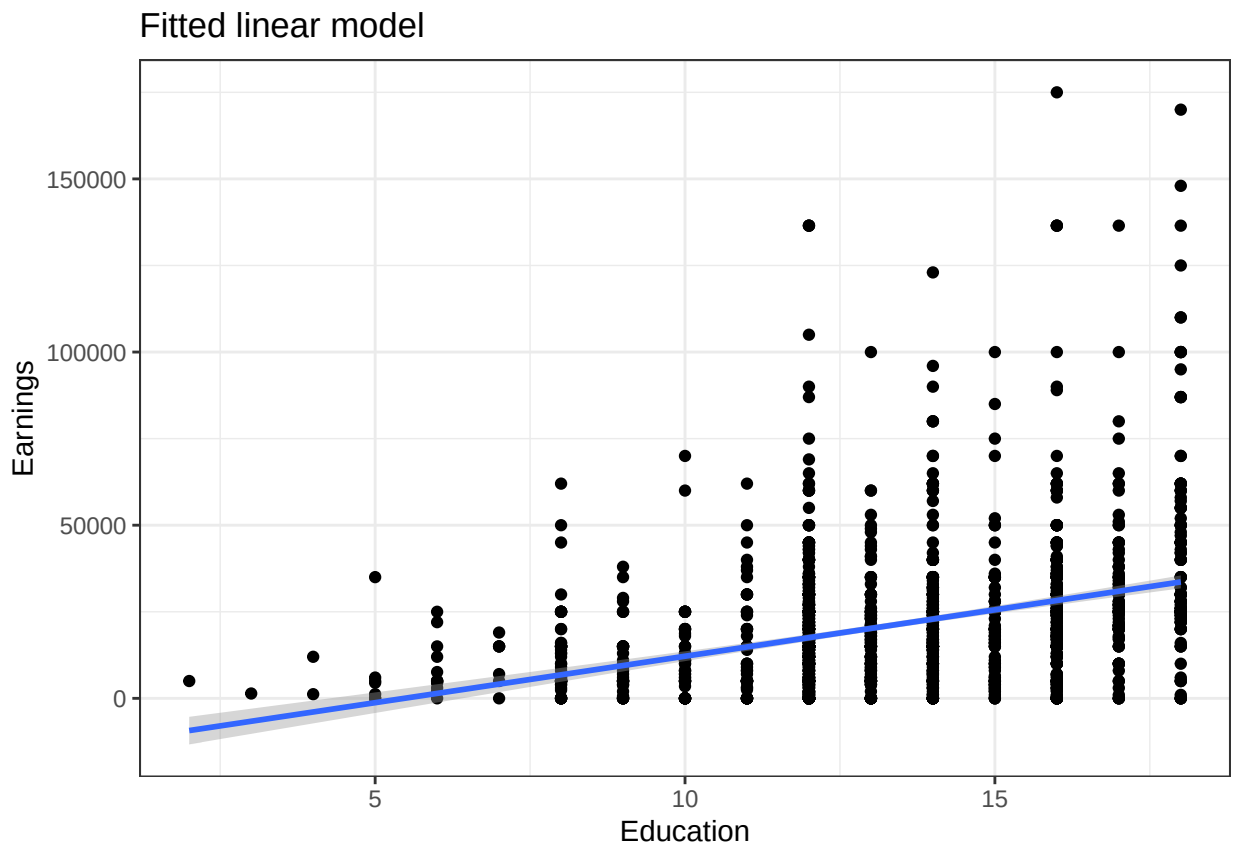
As we can see from the from the summary of the model output above, both the intercept and the slope are highly significant indicating that the *earn* variable is predicted well by the *education* variable.

Next using the subset of the with *earn* $<$ 2e+05, we can visualize how well our linear model explains *earn* using just the *education* covariate

```
earn_subset <- subset(df, df$earn < 2e5)

ggplot(earn_subset, aes(x=education, y=earn)) + geom_point(na.rm=TRUE) +   geom_smooth(method='lm', na.
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



(c) (20 points) Draw the qqplot by using the library ggplot for the model obtained in point b. Then perform the qqplot (using the library ggplot) for the two different groups of sex (similar to Figure 3 below). Take special care to the legend and the label. What can you say about this plot?

The code below creates a QQ plot on the residuals of the model above.

As we can see the residuals are in fact not normally distributed. Instead we see that there is a rightward skew such that the mode of the distribution is negative, but a larger proportion of the probability mass is spread over positive residuals

```
model_residuals <- resid(earn_fit)
p <- seq(0.05, 0.95, 0.01)
sample_quantiles <- quantile(model_residuals, p)
theoretical_quantiles <- qnorm(p, mean=mean(model_residuals), sd=sd(model_residuals))
qplot(theoretical_quantiles, sample_quantiles) + geom_abline() + labs(title='Residuals Q-Q Plot', y='Res
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
```

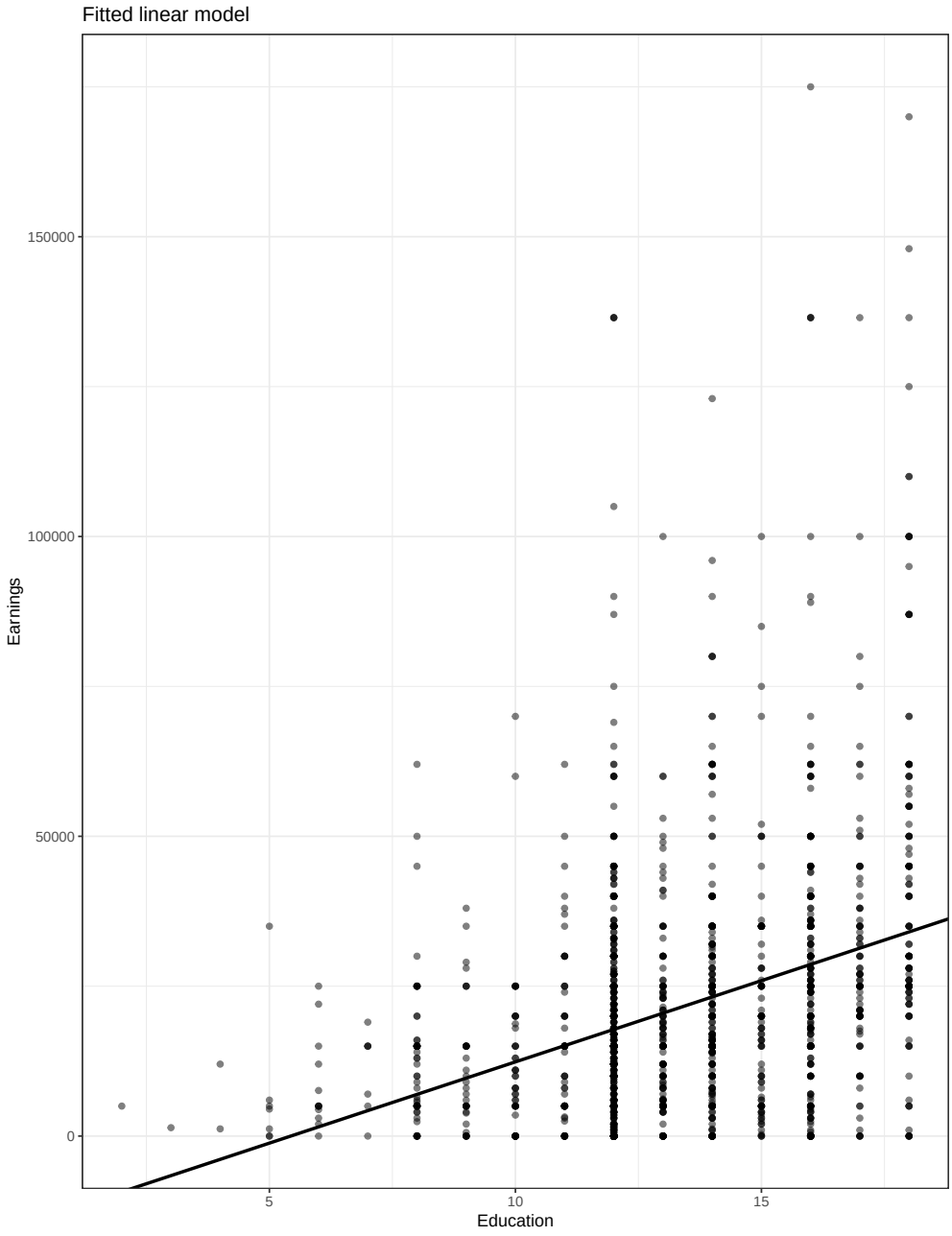
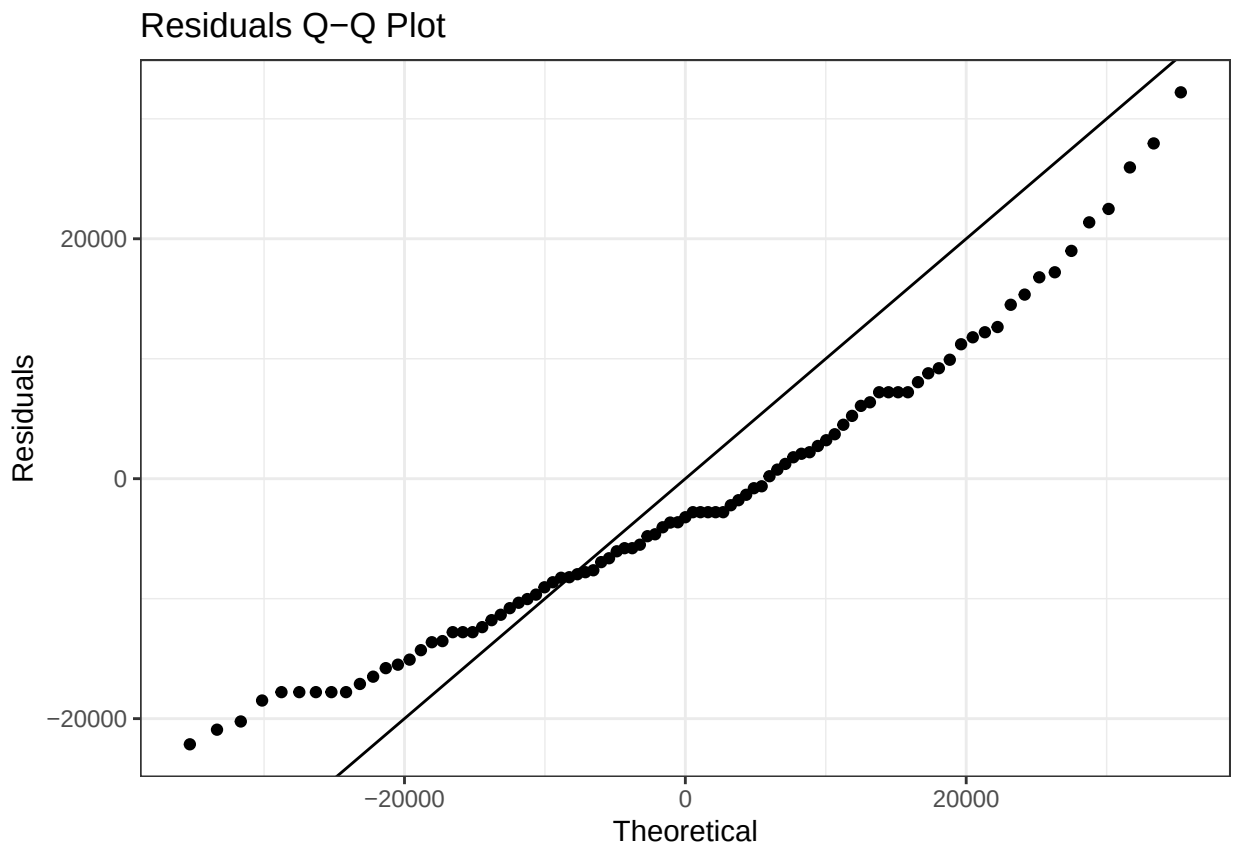


Figure 2: Linear Regression



Next we can visualize the QQ plots of the earn variable (not the residuals) for males and females separately. To allow for direct comparison the shows the quantiles with respect to a standard normal distribution. The first observation we can make is that neither of these subgroups have normally distributed earnings. Instead there is a strong positive skew such that the majority of males and females make a relatively small amount of money, but a smaller proportion with high earnings are spread out.

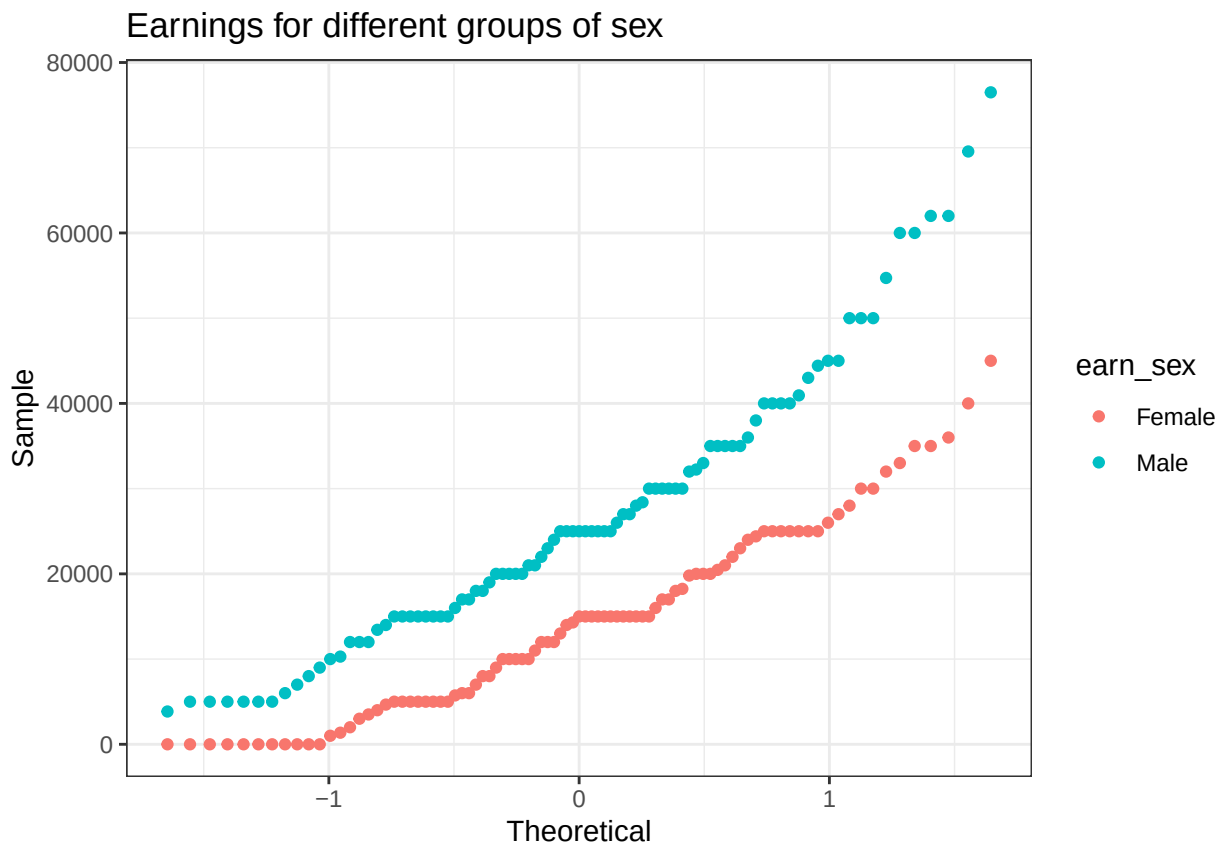
```
earn_quantiles_male <- quantile(subset(df, df$male==1)$earn, p, na.rm=TRUE)
earn_quantiles_female <- quantile(subset(df, df$male==0)$earn, p, na.rm=TRUE)

theoretical_quantiles_male <- qnorm(p, mean=0, sd=1)
theoretical_quantiles_female <- qnorm(p, mean=0, sd=1)

earn_quantiles <- c(earn_quantiles_male, earn_quantiles_female)
theoretical_quantiles <- c(theoretical_quantiles_male, theoretical_quantiles_female)
earn_sex <- c(rep('Male', each=length(earn_quantiles_male)),
              rep('Female', each=length(earn_quantiles_female)))

earn_sex_df <- data.frame(earn_quantiles, theoretical_quantiles, earn_sex)

ggplot(earn_sex_df, aes(x=theoretical_quantiles, y=earn_quantiles, color=earn_sex)) +
  geom_point() + labs(title='Earnings for different groups of sex', y='Sample', x='Theoretical') +
  theme_bw()
```



(d) (20 points) Perform in R the backward and forward procedure to select the covariates, remember to remove the rows with missing values. Did you obtain the same or different results from the two different procedures, please explain. Which procedure would you prefer? Comment what you discovered and the theoretical implications. Just for the backward solution compute the RSS and show the trend of RSS for beta1 in a plot by using ggplot in R (similar to Figure 4). (Hint: For RSS plot, set the range of x-axis to be [0,1000]).

The code below performs the backward procedure to select the best fit model. The procedure starts with a model that contains all covariates, and successively removes variables that have the lowest predictive ability.

```
fit1 <- lm(earn~., data=drop_na(df))
fit2 <- lm(earn~1, data=drop_na(df))
mod_backward <- stepAIC(fit1, direction="backward", scope=list(upper=fit1, lower=fit2))
```

```
## Start:  AIC=28708.32
## earn ~ height + weight + male + ethnicity + education + mother_education +
##     father_education + walk + exercise + smokenow + tense + angry +
##     age
##
##
## Step:  AIC=28708.32
## earn ~ height + weight + male + ethnicity + education + mother_education +
##     father_education + walk + exercise + smokenow + tense + age
##
##
## Step:  AIC=28708.32
## earn ~ height + weight + male + ethnicity + education + mother_education +
```

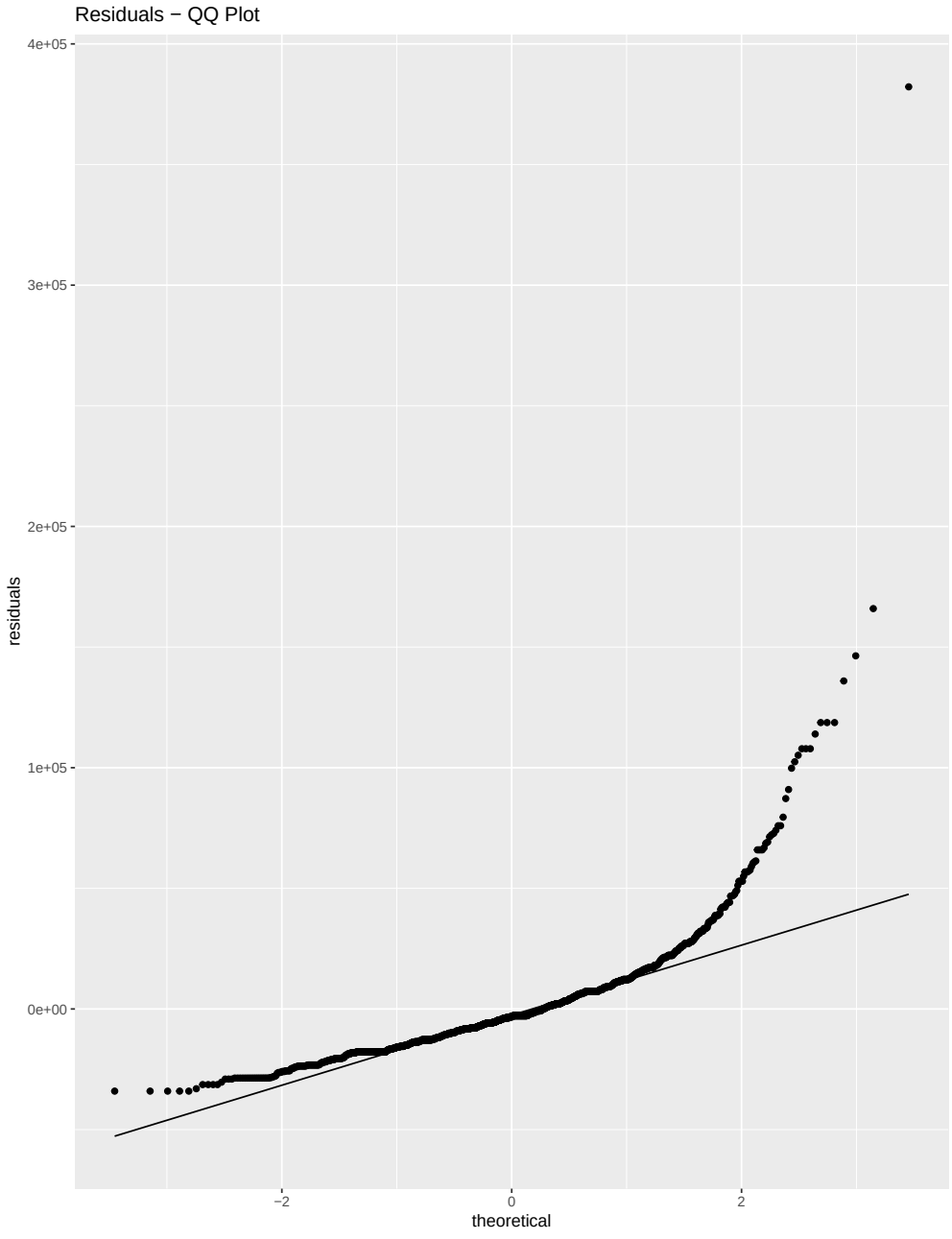


Figure 3: QQplot for different groups

```
##     walk + exercise + smokenow + tense + age
##
##
## Step:  AIC=28708.32
## earn ~ height + weight + male + ethnicity + education + walk +
##     exercise + smokenow + tense + age
##
##             Df  Sum of Sq        RSS   AIC
## - ethnicity  3 1.4940e+09 6.4531e+11 28706
## - walk       1 3.1861e+07 6.4385e+11 28706
## - weight     1 5.1199e+07 6.4387e+11 28706
## - smokenow   1 2.5692e+08 6.4407e+11 28707
## - exercise   1 3.2969e+08 6.4415e+11 28707
## <none>                    6.4382e+11 28708
## - tense      1 1.4955e+09 6.4531e+11 28710
## - height     1 2.2805e+09 6.4610e+11 28711
## - age        1 2.0675e+10 6.6449e+11 28752
## - male       1 2.5472e+10 6.6929e+11 28762
## - education  1 6.6279e+10 7.1009e+11 28847
##
## Step:  AIC=28705.66
## earn ~ height + weight + male + education + walk + exercise +
##     smokenow + tense + age
##
##             Df  Sum of Sq        RSS   AIC
## - walk       1 6.3440e+07 6.4537e+11 28704
## - weight     1 7.7849e+07 6.4539e+11 28704
## - smokenow   1 3.0394e+08 6.4561e+11 28704
## - exercise   1 3.4289e+08 6.4565e+11 28704
## <none>                    6.4531e+11 28706
## - tense      1 1.4071e+09 6.4672e+11 28707
## - height     1 2.8270e+09 6.4814e+11 28710
## - age        1 2.2170e+10 6.6748e+11 28752
## - male       1 2.4948e+10 6.7026e+11 28758
## - education  1 6.7532e+10 7.1284e+11 28847
##
## Step:  AIC=28703.8
## earn ~ height + weight + male + education + exercise + smokenow +
##     tense + age
##
##             Df  Sum of Sq        RSS   AIC
## - weight     1 7.4075e+07 6.4545e+11 28702
## - exercise   1 3.0181e+08 6.4568e+11 28702
## - smokenow   1 3.1560e+08 6.4569e+11 28702
## <none>                    6.4537e+11 28704
## - tense      1 1.4239e+09 6.4680e+11 28705
## - height     1 2.8131e+09 6.4819e+11 28708
## - age        1 2.2147e+10 6.6752e+11 28750
## - male       1 2.4931e+10 6.7030e+11 28756
## - education  1 6.7475e+10 7.1285e+11 28845
##
## Step:  AIC=28701.97
## earn ~ height + male + education + exercise + smokenow + tense +
##     age
```

```
##
##             Df  Sum of Sq        RSS   AIC
## - smokenow   1 3.3784e+08 6.4579e+11 28701
## - exercise   1 3.4429e+08 6.4579e+11 28701
## <none>                    6.4545e+11 28702
## - tense      1 1.4060e+09 6.4685e+11 28703
## - height     1 2.9325e+09 6.4838e+11 28706
## - age        1 2.2336e+10 6.6778e+11 28749
## - male       1 2.5233e+10 6.7068e+11 28755
## - education  1 6.7691e+10 7.1314e+11 28844
##
## Step:  AIC=28700.72
## earn ~ height + male + education + exercise + tense + age
##
##             Df  Sum of Sq        RSS   AIC
## - exercise   1 2.8870e+08 6.4607e+11 28699
## <none>                    6.4579e+11 28701
## - tense      1 1.4833e+09 6.4727e+11 28702
## - height     1 2.9449e+09 6.4873e+11 28705
## - age        1 2.2071e+10 6.6786e+11 28747
## - male       1 2.5335e+10 6.7112e+11 28754
## - education  1 6.7697e+10 7.1348e+11 28842
##
## Step:  AIC=28699.37
## earn ~ height + male + education + tense + age
##
##             Df  Sum of Sq        RSS   AIC
## <none>                    6.4607e+11 28699
## - tense      1 1.4303e+09 6.4750e+11 28700
## - height     1 3.0014e+09 6.4908e+11 28704
## - age        1 2.2443e+10 6.6852e+11 28746
## - male       1 2.6396e+10 6.7247e+11 28755
## - education  1 6.9666e+10 7.1574e+11 28845
```

```
summary(mod_backward)
```

```
##
## Call:
## lm(formula = earn ~ height + male + education + tense + age,
##     data = drop_na(df))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43220 -10972  -2314   6306 371527
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -70750.77   14208.16  -4.980 7.14e-07 ***
## height         550.82     213.41   2.581  0.00995 **
## male         12694.11    1658.43   7.654 3.55e-14 ***
## education     2952.42     237.43  12.435  < 2e-16 ***
## tense          490.96     275.55   1.782  0.07500 .
## age            257.45      36.48   7.058 2.62e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 21230 on 1434 degrees of freedom
## Multiple R-squared:  0.2116, Adjusted R-squared:  0.2089
## F-statistic: 76.98 on 5 and 1434 DF,  p-value: < 2.2e-16
```

Likewise, we can perform the forward pass by starting with a minimal model that predicts a flat line, and successively add the covariates which individually have the highest predictive ability (lowest RSS when they are used as the solve covariate).

```
mod_forward <- stepAIC(fit2, direction="forward", scope=list(upper=fit1, lower=fit2))
```

```
## Start:  AIC=29031.73
## earn ~ 1
##
##                     Df  Sum of Sq        RSS   AIC
## + male               1 7.9739e+10 7.3974e+11 28886
## + education          1 7.4527e+10 7.4495e+11 28896
## + mother_education   1 7.4527e+10 7.4495e+11 28896
## + father_education   1 7.4527e+10 7.4495e+11 28896
## + height             1 5.9505e+10 7.5997e+11 28925
## + weight             1 2.9325e+10 7.9016e+11 28981
## + age                1 1.3370e+10 8.0611e+11 29010
## + exercise           1 5.0787e+09 8.1440e+11 29025
## + tense              1 3.9410e+09 8.1554e+11 29027
## + angry              1 3.9410e+09 8.1554e+11 29027
## + ethnicity          3 4.8139e+09 8.1467e+11 29029
## <none>                             8.1948e+11 29032
## + smokenow           1 9.1854e+08 8.1856e+11 29032
## + walk               1 9.0848e+08 8.1857e+11 29032
##
## Step:  AIC=28886.32
## earn ~ male
##
##                     Df  Sum of Sq        RSS   AIC
## + education          1 6.9797e+10 6.6994e+11 28746
## + mother_education   1 6.9797e+10 6.6994e+11 28746
## + father_education   1 6.9797e+10 6.6994e+11 28746
## + age                1 1.7756e+10 7.2198e+11 28853
## + height             1 3.6894e+09 7.3605e+11 28881
## + ethnicity          3 5.3068e+09 7.3443e+11 28882
## <none>                             7.3974e+11 28886
## + weight             1 9.4776e+08 7.3879e+11 28886
## + smokenow           1 9.1854e+08 7.3882e+11 28886
## + tense              1 3.4937e+08 7.3939e+11 28888
## + angry              1 3.4937e+08 7.3939e+11 28888
## + walk               1 2.3203e+08 7.3951e+11 28888
## + exercise           1 9.0739e+07 7.3965e+11 28888
##
## Step:  AIC=28745.61
## earn ~ male + education
##
##              Df  Sum of Sq        RSS   AIC
## + age         1 1.9518e+10 6.5043e+11 28705
## + ethnicity   3 3.5226e+09 6.6642e+11 28744
## + height      1 1.4225e+09 6.6852e+11 28744
```

```
## + weight     1 1.0396e+09 6.6890e+11 28745
## <none>                    6.6994e+11 28746
## + exercise   1 5.4813e+08 6.6940e+11 28746
## + smokenow   1 1.1241e+08 6.6983e+11 28747
## + walk       1 3.1565e+06 6.6994e+11 28748
## + tense      1 2.5823e+06 6.6994e+11 28748
## + angry      1 2.5823e+06 6.6994e+11 28748
##
## Step:  AIC=28705.03
## earn ~ male + education + age
##
##             Df  Sum of Sq        RSS   AIC
## + height     1 2921837319 6.4750e+11 28700
## + tense      1 1350718589 6.4908e+11 28704
## + angry      1 1350718589 6.4908e+11 28704
## <none>                    6.5043e+11 28705
## + smokenow   1  362030699 6.5006e+11 28706
## + ethnicity  3 2093480782 6.4833e+11 28706
## + exercise   1  287699226 6.5014e+11 28706
## + weight     1  137467965 6.5029e+11 28707
## + walk       1   33481537 6.5039e+11 28707
##
## Step:  AIC=28700.55
## earn ~ male + education + age + height
##
##             Df  Sum of Sq        RSS   AIC
## + tense      1 1430271509 6.4607e+11 28699
## + angry      1 1430271509 6.4607e+11 28699
## <none>                    6.4750e+11 28700
## + smokenow   1  357020068 6.4715e+11 28702
## + exercise   1  235648665 6.4727e+11 28702
## + weight     1  111208434 6.4739e+11 28702
## + walk       1   41917712 6.4746e+11 28702
## + ethnicity  3 1521906378 6.4598e+11 28703
##
## Step:  AIC=28699.37
## earn ~ male + education + age + height + tense
##
##             Df  Sum of Sq        RSS   AIC
## <none>                    6.4607e+11 28699
## + exercise   1  288700139 6.4579e+11 28701
## + smokenow   1  282249180 6.4579e+11 28701
## + weight     1  137259702 6.4594e+11 28701
## + walk       1   25908265 6.4605e+11 28701
## + ethnicity  3 1607572422 6.4447e+11 28702
```

```
summary(mod_forward)
```

```
##
## Call:
## lm(formula = earn ~ male + education + age + height + tense,
##     data = drop_na(df))
##
## Residuals:
##    Min     1Q Median     3Q    Max
```

```
## -43220 -10972  -2314   6306 371527
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -70750.77   14208.16  -4.980 7.14e-07 ***
## male         12694.11    1658.43   7.654 3.55e-14 ***
## education     2952.42     237.43  12.435  < 2e-16 ***
## age            257.45      36.48   7.058 2.62e-12 ***
## height         550.82     213.41   2.581  0.00995 **
## tense          490.96     275.55   1.782  0.07500 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21230 on 1434 degrees of freedom
## Multiple R-squared:  0.2116, Adjusted R-squared:  0.2089
## F-statistic: 76.98 on 5 and 1434 DF,  p-value: < 2.2e-16
```

From the results above we can see that the forward and backward passes actually arrive at the same final model of 'earn ~ height + male + education + tense + age'. Since the both approaches pick the optimal beta coefficients for these terms, the models are in fact identical.

To find the RSS associated with changing the beta 1 coefficient, we will first create a function that manually performs the matrix multiplication using the weights arrived at from the backward pass, and then updates the coefficient based on our input. With this function we can simply pass in the true beta coefficient for the height variable and see that the resulting RSS is 6.46e+11

```
beta_update <- function(new_val, mod, d){
  beta_coef <- mod[["coefficients"]]
  beta_coef[2] <- new_val
  mod_d <- subset(d, select=names(mod[["coefficients"]][c(2,3,4,5,6)]))
  mod_d <- data.frame(intercept=rep(1, nrow(mod_d)), mod_d)
  pred <- as.matrix(mod_d) %*% beta_coef

  res <- sum((d$earn - pred)^2)
  return(res)
}

beta_update(mod_backward[["coefficients"]][2], mod_backward, drop_na(df))
```

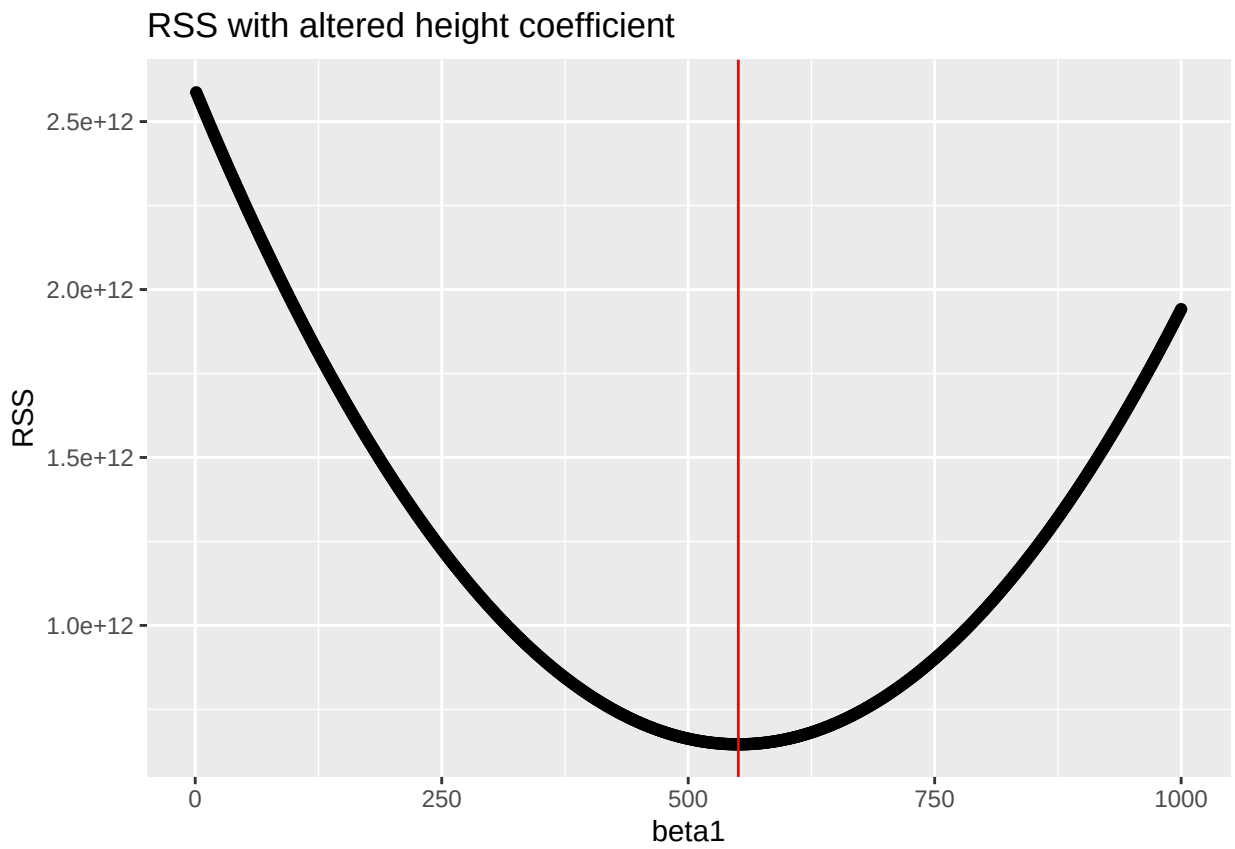
```
## [1] 6.46074e+11
```

Using this function in a for loop we can plot the RSS as a function of the beta 1 coefficient and observe that the minima occurs exactly at the beta coefficient used in the true model (red line).

```
rss_out <- rep(0, 1000)
for (i in 1:1000){
  rss_out[i] <- beta_update(i, mod_backward, drop_na(df))
}

rss_df <- data.frame(RSS=rss_out, beta1=c(1:1000))

ggplot(data=rss_df, aes(x=beta1, y=rss_out)) + geom_point() +
  labs(title="RSS with altered height coefficient", y="RSS") +
  geom_vline(xintercept=mod_backward[["coefficients"]][2], color="red")
```



(e) (20 points) Perform a bootstrap of 500 samples for beta 1 (*height*), beta 2 (*male*), and beta 3 (*education*) for the coefficient obtained in the backward procedure in point d. Plot the beta coefficients that you have obtained with histograms with ggplot (similar to Figure 5). Remember to use the data without missing values.

The code below creates a function that fits a linear model to a subset of the earnings dataset, and returns the resulting beta coefficients. Combining this with bootstrap function, we can aggregate the beta coefficients for randoms sets of 500.

```
fc <- function(d, i){
  d2 <- d[i,]
  boot_fit <- lm(earn ~ height + male + education + tense + age, data=d2)

  return(boot_fit[['coefficients']])
}

earn_boot <- boot(drop_na(df), fc, R=500)
```

Next we can plot the beta coefficients from the bootstrap samples as a histogram and plot them against the true beta coefficients derived from a linear model on the full earnings dataset.

```
earn_boot_coef <- data.frame(earn_boot[["t"]])
colnames(earn_boot_coef) <- names(earn_boot[["t0"]])

earn_boot_coef <- subset(earn_boot_coef, select=c('height', 'male', 'education'))
earn_boot_coef <- gather(earn_boot_coef)
earn_boot_vline <- data.frame(key=c('height', 'male', 'education'),
                              value=c(earn_boot[["t0"]][["height"]],
```

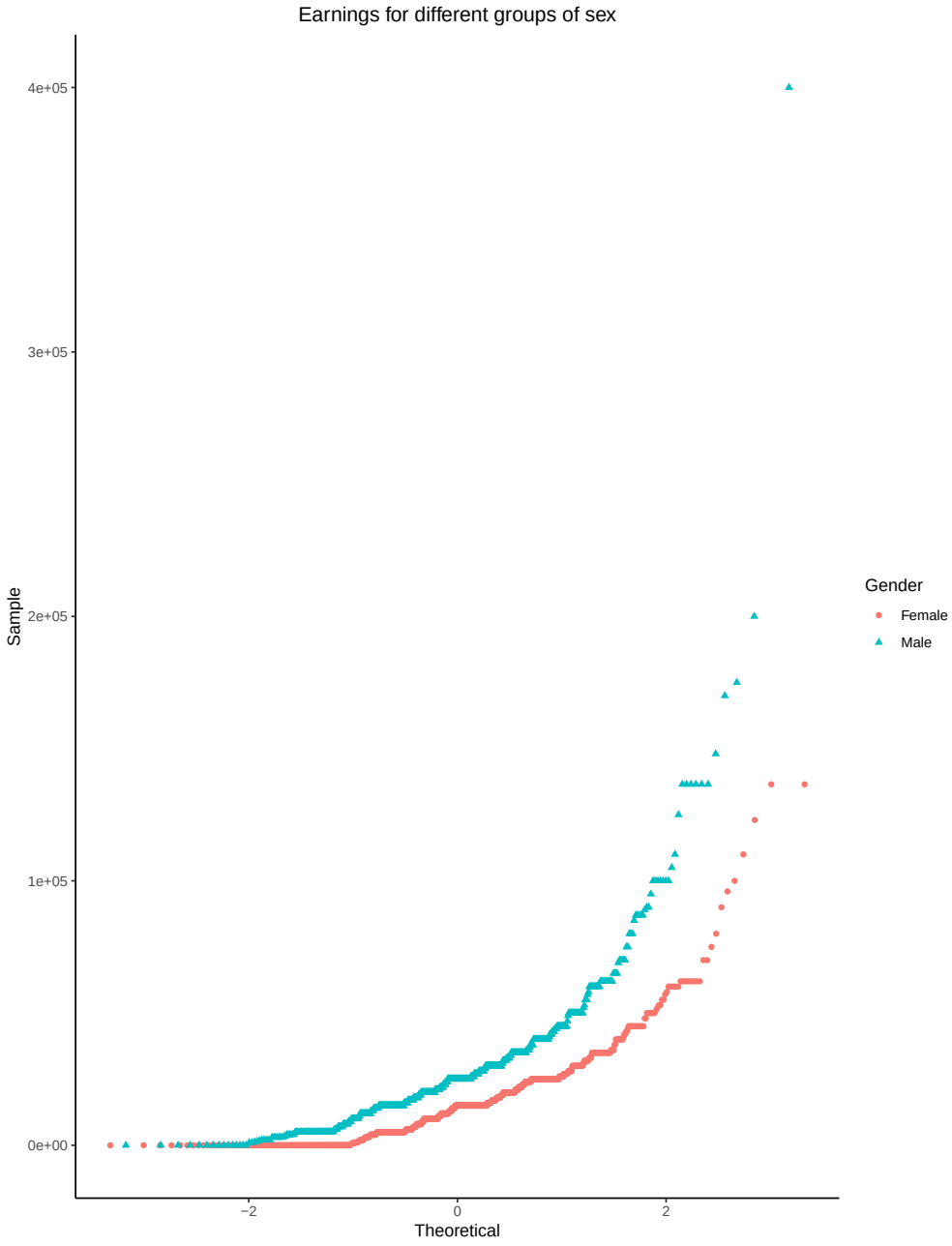
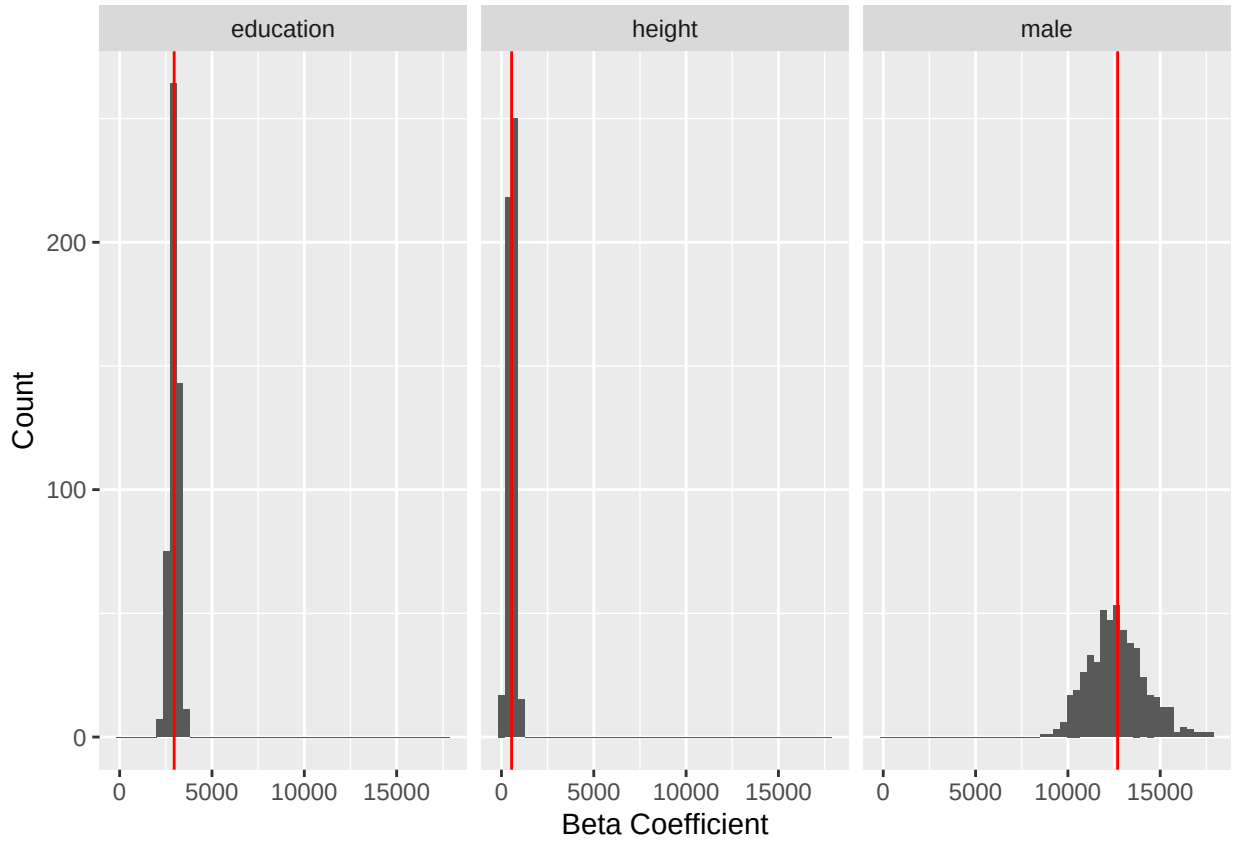


Figure 4: RSS for the backward procedure

```
                                    earn_boot[["t0"]][["male"]],
                                    earn_boot[["t0"]][["education"]]))

ggplot(earn_boot_coef, aes(x=value)) + geom_histogram(bins=50) +
  labs(x='Beta Coefficient', y='Count') +
  geom_vline(data=earn_boot_vline, aes(xintercept=value), color="red") +
  facet_wrap(~ key)
```




```
  theme_bw()
```

```
## List of 94
##  $ line                      :List of 6
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                      :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
```

```
##  $ text                        :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                       : NULL
##  $ aspect.ratio                : NULL
##  $ axis.title                  : NULL
##  $ axis.title.x                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75pt 0pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top            :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 2.75pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom         : NULL
##  $ axis.title.y                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
```

```
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 2.75pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left         : NULL
##  $ axis.title.y.right        :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 0pt 2.75pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2pt 0pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
```

```
##  $ axis.text.x.top           :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 2.2pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom        : NULL
##  $ axis.text.y               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 2.2pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left          : NULL
##  $ axis.text.y.right         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 0pt 2.2pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                :List of 6
##   ..$ colour       : chr "grey20"
##   ..$ linewidth    : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
```

```
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ axis.ticks.x              : NULL
##  $ axis.ticks.x.top          : NULL
##  $ axis.ticks.x.bottom       : NULL
##  $ axis.ticks.y              : NULL
##  $ axis.ticks.y.left         : NULL
##  $ axis.ticks.y.right        : NULL
##  $ axis.ticks.length         : 'unit' num 2.75pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ axis.ticks.length.x       : NULL
##  $ axis.ticks.length.x.top   : NULL
##  $ axis.ticks.length.x.bottom: NULL
##  $ axis.ticks.length.y       : NULL
##  $ axis.ticks.length.y.left  : NULL
##  $ axis.ticks.length.y.right : NULL
##  $ axis.line                 : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x               : NULL
##  $ axis.line.x.top           : NULL
##  $ axis.line.x.bottom        : NULL
##  $ axis.line.y               : NULL
##  $ axis.line.y.left          : NULL
##  $ axis.line.y.right         : NULL
##  $ legend.background         :List of 5
##   ..$ fill         : NULL
##   ..$ colour       : logi NA
##   ..$ linewidth    : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ legend.margin             : 'margin' num [1:4] 5.5pt 5.5pt 5.5pt 5.5pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ legend.spacing            : 'unit' num 11pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ legend.spacing.x          : NULL
##  $ legend.spacing.y          : NULL
##  $ legend.key                :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : logi NA
##   ..$ linewidth    : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ legend.key.size           : 'unit' num 1.2lines
##   ..- attr(*, "valid.unit")= int 3
##   ..- attr(*, "unit")= chr "lines"
##  $ legend.key.height         : NULL
##  $ legend.key.width          : NULL
##  $ legend.text               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
```

```
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.align         : NULL
##  $ legend.title              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.align        : NULL
##  $ legend.position           : chr "right"
##  $ legend.direction          : NULL
##  $ legend.justification      : chr "center"
##  $ legend.box                : NULL
##  $ legend.box.just           : NULL
##  $ legend.box.margin         : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "valid.unit")= int 1
##   ..- attr(*, "unit")= chr "cm"
##  $ legend.box.background     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing        : 'unit' num 11pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ panel.background          :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : logi NA
##   ..$ linewidth    : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ panel.border              :List of 5
##   ..$ fill         : logi NA
##   ..$ colour       : chr "grey20"
##   ..$ linewidth    : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ panel.spacing             : 'unit' num 5.5pt
##   ..- attr(*, "valid.unit")= int 8
```

```
##   ..- attr(*, "unit")= chr "pt"
##  $ panel.spacing.x           : NULL
##  $ panel.spacing.y           : NULL
##  $ panel.grid                :List of 6
##   ..$ colour       : chr "grey92"
##   ..$ linewidth    : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major          : NULL
##  $ panel.grid.minor          :List of 6
##   ..$ colour       : NULL
##   ..$ linewidth    : 'rel' num 0.5
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major.x        : NULL
##  $ panel.grid.major.y        : NULL
##  $ panel.grid.minor.x        : NULL
##  $ panel.grid.minor.y        : NULL
##  $ panel.ontop               : logi FALSE
##  $ plot.background           :List of 5
##   ..$ fill         : NULL
##   ..$ colour       : chr "white"
##   ..$ linewidth    : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ plot.title                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 5.5pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.title.position       : chr "panel"
##  $ plot.subtitle             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
```

```
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0pt 0pt 5.5pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : num 1
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 5.5pt 0pt 0pt 0pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption.position     : chr "panel"
##  $ plot.tag                  :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.tag.position         : chr "topleft"
##  $ plot.margin               : 'margin' num [1:4] 5.5pt 5.5pt 5.5pt 5.5pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ strip.background          :List of 5
##   ..$ fill         : chr "grey85"
##   ..$ colour       : chr "grey20"
##   ..$ linewidth    : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ strip.background.x        : NULL
##  $ strip.background.y        : NULL
##  $ strip.clip                : chr "inherit"
##  $ strip.placement           : chr "inside"
##  $ strip.text                :List of 11
```

```
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey10"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 4.4pt 4.4pt 4.4pt 4.4pt
##   .. ..- attr(*, "valid.unit")= int 8
##   .. ..- attr(*, "unit")= chr "pt"
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.x              : NULL
##  $ strip.text.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.switch.pad.grid     : 'unit' num 2.75pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ strip.switch.pad.wrap     : 'unit' num 2.75pt
##   ..- attr(*, "valid.unit")= int 8
##   ..- attr(*, "unit")= chr "pt"
##  $ strip.text.y.left         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

(f) (20 points) Compute the LOO and K-fold cross validation and write the results. Compute the mean square error for both the LOO and the K-fold cross validation. Then plot the prediction against the true value for LOO, using ggplot. Describe the results. Remember to use the data without missing

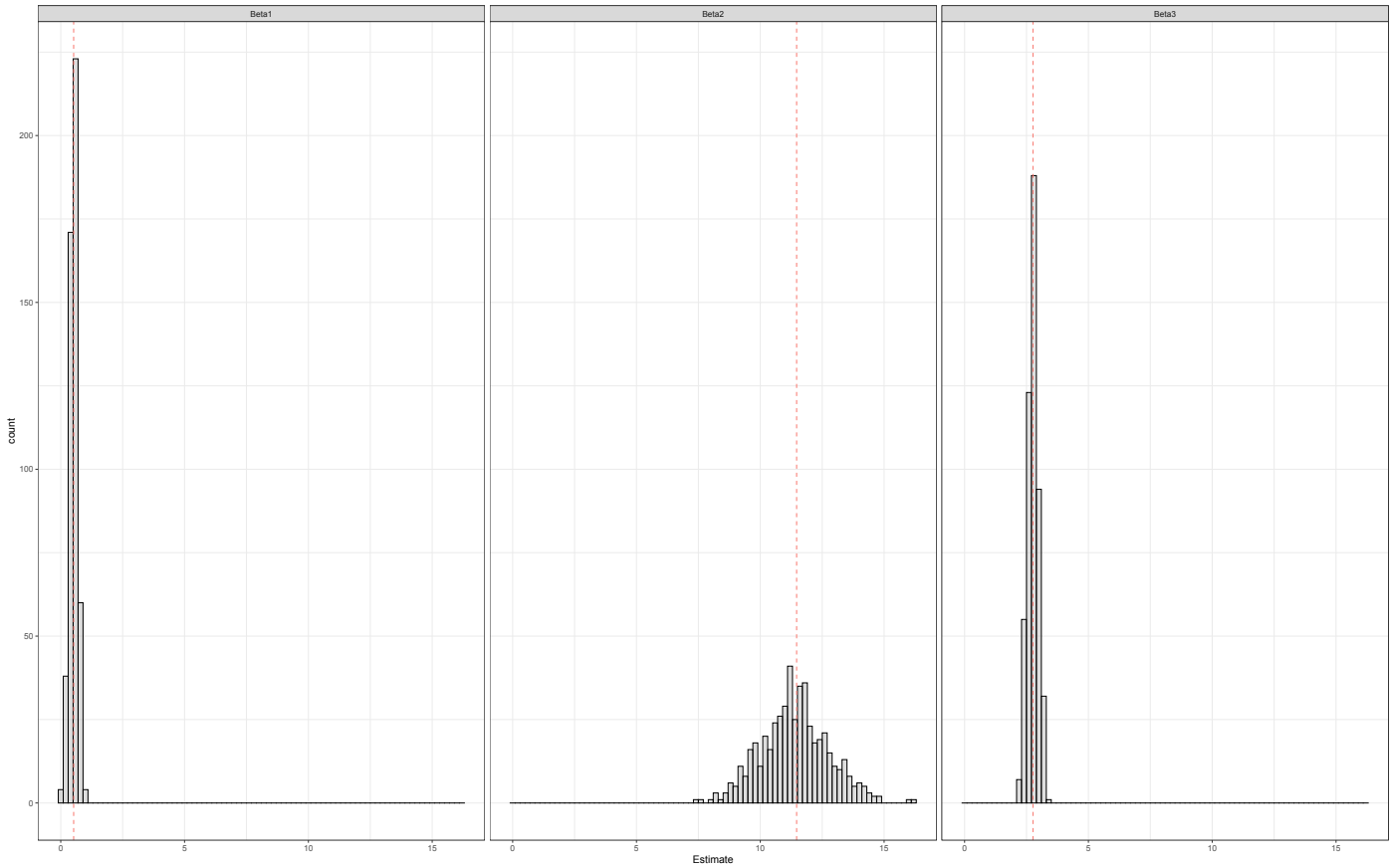


Figure 5: Bootstrap Results

values.

The code below performs leave-one-out cross validation using the model arrived at from the forward and backward passes. The RMSE calculated using this procedure is 21,289.3, or an MSE of 453,234,294.

```
train.control <- trainControl(method = "LOOCV")
model_loocv <- train(earn~height + male + education + tense + age, data=drop_na(df),
               method="lm", trControl=train.control)
print(model_loocv)
```

```
## Linear Regression
##
## 1440 samples
##    5 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 1439, 1439, 1439, 1439, 1439, 1439, ...
## Resampling results:
##
##   RMSE     Rsquared   MAE
##   21289.3  0.2036473  12870.92
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
summary(model_loocv)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43220 -10972  -2314   6306 371527
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -70750.77   14208.16  -4.980 7.14e-07 ***
## height         550.82     213.41   2.581  0.00995 **
## male         12694.11    1658.43   7.654 3.55e-14 ***
## education     2952.42     237.43  12.435  < 2e-16 ***
## tense          490.96     275.55   1.782  0.07500 .
## age            257.45      36.48   7.058 2.62e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21230 on 1434 degrees of freedom
## Multiple R-squared:  0.2116, Adjusted R-squared:  0.2089
## F-statistic: 76.98 on 5 and 1434 DF,  p-value: < 2.2e-16
```

The code below performs the same steps for 10 fold cross validation. We see that the RMSE from this procedure is 20,572.79, or an MSE of 423,239,688

```
train.control <- trainControl(method = "cv", number=10)
model_cv <- train(earn~height + male + education + tense + age, data=drop_na(df),
               method="lm", trControl=train.control)
print(model_cv)
```

```
## Linear Regression
##
## 1440 samples
##    5 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1296, 1294, 1296, 1297, 1297, 1296, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   20660.78  0.2298471  12892.72
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
summary(model_cv)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43220 -10972  -2314   6306 371527
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -70750.77   14208.16  -4.980 7.14e-07 ***
## height         550.82     213.41   2.581  0.00995 **
## male         12694.11    1658.43   7.654 3.55e-14 ***
## education     2952.42     237.43  12.435  < 2e-16 ***
## tense          490.96     275.55   1.782  0.07500 .
## age            257.45      36.48   7.058 2.62e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21230 on 1434 degrees of freedom
## Multiple R-squared:  0.2116, Adjusted R-squared:  0.2089
## F-statistic: 76.98 on 5 and 1434 DF,  p-value: < 2.2e-16
```

```
model_summ_cv <-summary(model_cv[["finalModel"]])
mean(model_summ_cv$residuals^2)
```

```
## [1] 448662467
```

Comparing the results of the two cross validation procedures, we see that the estimated MSE is slightly smaller using 10 fold cross validation. As expected from the MSE, we additionally see the estimated R squared from 10 fold CV is higher for LOOCV (0.24 vs 0.2). Despite these minor differences it can be concluded that the procedures produce largely similar results for the efficacy this model for predicting earnings.

The code below plots the true vs the predicted earnings using the model from the LOOCV procedure. As we can see the predictions are concentrated around the diagonal which indicates perfect accuracy. However there are several outliers where the true earnings for certain individuals are predicted to be much lower than reality. Additionally we see that some individuals are predicted to have negative earnings which is clearly unrealistic since the minimum earnings is zero.

```
pred <- predict(model_loocv, newdata = drop_na(df))
cv_df <- data.frame(true_earn=drop_na(df)$earn, pred_earn=pred)
ggplot(cv_df, aes(x=true_earn, y=pred_earn)) + geom_point() +
  labs(title='Linear Model True Earnings vs Predicted Earnings', x='True', y='Predicted') + theme_bw() +
```