HW6, SOLUTIONS

Problem #1

Code below:

```
library(ISLR)

lm.full <- lm(mpg ~ .-name,
              data=Auto)
summary(lm.full)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

1. We have 7 predictors, hence 7 slope parameters:

   $$H_0 : \beta_1 = \beta_2 = \cdots = \beta_7 = 0, \quad vs \quad H_a : \{\text{at least one } \beta_j \neq 0\}$$

   The very bottom lines describes the corresponding F-test results. The model is significant due to a tiny p-value (≈ 0), hence we reject H_0.

2. Displacement, weight, year, origin.

3. - *weight* (p-value ≈ 0): Per 1lb increase in weight, **holding all other predictors constant**, the car's miles per gallon will drop by 0.0064 miles, **on average**.

```r
confint(lm.full)[c(5),]
```

```
##        2.5 %       97.5 %
## -0.007756074 -0.005192013
```

4. - 95% CI for *weight*: $(-0.008, -0.005)$. Interpretation: We are 95% confident that per 1lb increase in weight, **holding other predictors constant**, the miles per gallon will decrease by between 0.005 and 0.008 miles, **on average**.
5. - $RSE = 3.328$: Our model's predicted miles per gallon miss the true values by 3.328mpg, **on average**.
     - $R^2 = 0.8215$: Our linear model, including 7 car characteristics, explains 82.15% of variability in miles per gallon.

Problem #2

1. Fitting the model:

```r
library(ISwR)
lm.obj <- lm(pemax ~ ., data=cystfibr)
summary(lm.obj)
```

```
##
## Call:
## lm(formula = pemax ~ ., data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.338 -11.532   1.081  13.386  33.405
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 176.0582   225.8912   0.779    0.448
## age          -2.5420     4.8017  -0.529    0.604
## sex          -3.7368    15.4598  -0.242    0.812
## height       -0.4463     0.9034  -0.494    0.628
## weight        2.9928     2.0080   1.490    0.157
## bmp          -1.7449     1.1552  -1.510    0.152
## fev1          1.0807     1.0809   1.000    0.333
## rv            0.1970     0.1962   1.004    0.331
## frc          -0.3084     0.4924  -0.626    0.540
## tlc           0.1886     0.4997   0.377    0.711
##
## Residual standard error: 25.47 on 15 degrees of freedom
## Multiple R-squared:  0.6373, Adjusted R-squared:  0.4197
## F-statistic: 2.929 on 9 and 15 DF,  p-value: 0.03195
```

a. The overall model is significant due to a small p-value for F-test (0.03), while none of the individual predictors show up as significant (smallest p-value is 0.15). This is happening due to collinearity: some predictors are strongly correlated/collinear with each other, making it tougher to estimate their partial effects, hence **inflating the standard errors of estimates**.

b. Correlation matrix:

```
print(round(cor(cystfibr[,-10]),2))
```

```
##          age   sex height weight   bmp  fev1    rv   frc   tlc
## age     1.00 -0.17   0.93   0.91  0.38  0.29 -0.55 -0.64 -0.47
## sex    -0.17  1.00  -0.17  -0.19 -0.14 -0.53  0.27  0.18  0.02
## height  0.93 -0.17   1.00   0.92  0.44  0.32 -0.57 -0.62 -0.46
## weight  0.91 -0.19   0.92   1.00  0.67  0.45 -0.62 -0.62 -0.42
## bmp     0.38 -0.14   0.44   0.67  1.00  0.55 -0.58 -0.43 -0.36
## fev1    0.29 -0.53   0.32   0.45  0.55  1.00 -0.67 -0.67 -0.44
## rv     -0.55  0.27  -0.57  -0.62 -0.58 -0.67  1.00  0.91  0.59
## frc    -0.64  0.18  -0.62  -0.62 -0.43 -0.67  0.91  1.00  0.70
## tlc    -0.47  0.02  -0.46  -0.42 -0.36 -0.44  0.59  0.70  1.00
```

There's clear evidence of collinearity due to strong correlations among the following predictor groups:

- height, weight and age (all pairwise correlations are > 0.90)
- $fev1$ and frc ($cor(fev1, frc) = 0.91$)

For each group of strongly correlated predictors, we need to retain **only one** (sort of the "representative" for that group). That would lead to each feature in our final model to represent an **independent piece** of information.

Which ones to retain, which ones to drop? That's typically arbitrary. From the first group, let's retain *height* (hence dropping *weight* and *age*), from second - $fev1$ (dropping frc).

```
lm.obj <- lm(pemax ~ .-weight-age-frc, data=cystfibr)
summary(lm.obj)
```

```
##
## Call:
## lm(formula = pemax ~ . - weight - age - frc, data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.150 -19.032   3.555  14.318  42.201
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -166.4898    99.0066  -1.682  0.10991
## sex            4.2738    13.2334   0.323  0.75045
## height         1.1668     0.3183   3.666  0.00177 **
## bmp           -0.5629     0.5876  -0.958  0.35073
## fev1           1.9037     0.8003   2.379  0.02865 *
## rv             0.1095     0.1034   1.059  0.30366
## tlc            0.3990     0.4118   0.969  0.34544
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 25.96 on 18 degrees of freedom
## Multiple R-squared:  0.5479, Adjusted R-squared:  0.3971
## F-statistic: 3.635 on 6 and 18 DF,  p-value: 0.01535
```

Now we see *height* and *fev*1 showing up as statistically significant predictors (p-values of 0.002 and 0.03, respectively).

c. Code with clarifying comments below:

```
library(car)
```

```
## Loading required package: carData
```

```
lm.obj <- lm(pemax ~ ., data=cystfibr)
# Weight is the one with highest VIF of 47.78 (and it's >5), we drop it first.
vif(lm.obj)
```

```
##       age       sex    height    weight       bmp      fev1        rv       frc
## 21.829841  2.269407 13.954929 47.781303  7.115752  5.419507 10.538052 17.143073
##       tlc
##  2.659993
```

```
lm.obj <- lm(pemax ~ .-weight, data=cystfibr)

# frc is the one with highest VIF of 15.81 (and it's >5),  we drop it here.
vif(lm.obj)
```

```
##       age       sex    height       bmp      fev1        rv       frc       tlc
##  8.097571  2.029182  7.595539  2.730462  4.205260 10.332505 15.814231  2.177076
```

```
lm.obj <- lm(pemax ~ .-weight-frc, data=cystfibr)

# height is the one with highest VIF of 7.59 (and it's >5), hence we drop it here.
vif(lm.obj)
```

```
##      age      sex   height      bmp     fev1       rv      tlc
## 7.341695 1.606561 7.595520 1.794168 2.870202 2.836471 1.768577
```

```
lm.obj <- lm(pemax ~ .-weight-frc-height, data=cystfibr)

# No VIF is higher than 5 => no evidence of collinearity => stop here.
vif(lm.obj)
```

```
##      age      sex      bmp     fev1       rv      tlc
## 1.611582 1.605444 1.718765 2.861478 2.814628 1.768466
```

d. We have to estimate **partial effects** of each predictor - "per 1-unit increase in predictor x, holding **other predictors constant**". If some predictors are collinear, we don't see enough examples of one of them increasing, while the other ones are constant. E.g. if x_1 and x_2 are highly correlated, and we see an increase in x_1, x_2 also changes with it (and vice versa), making the estimation of $\hat{\beta}_1, \hat{\beta}_2$ unreliable $\implies$ variance/std. errors of $\hat{\beta}_1, \hat{\beta}_2$ gets inflated.

2. Fitting the models:

```r
library(ISwR)
summary(lm(pemax ~ sex + height + rv, data=cystfibr))
```

```
##
## Call:
## lm(formula = pemax ~ sex + height + rv, data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.998 -17.998   0.313  21.685  55.725
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -40.56525   62.97033  -0.644  0.52642
## sex         -13.88720   11.58232  -1.199  0.24388
## height        0.95932    0.31958   3.002  0.00679 **
## rv            0.03609    0.08182   0.441  0.66362
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.66 on 21 degrees of freedom
## Multiple R-squared:  0.401,  Adjusted R-squared:  0.3155
## F-statistic: 4.687 on 3 and 21 DF,  p-value: 0.0117
```

```r
summary(lm(pemax ~ sex + weight + height + rv + frc, data=cystfibr))
```

```
##
## Call:
## lm(formula = pemax ~ sex + weight + height + rv + frc, data = cystfibr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -48.28 -19.82   0.91  15.83  37.97
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 104.4139    88.0781   1.185   0.2504
## sex         -16.6162    11.0070  -1.510   0.1476
## weight        1.5027     0.8211   1.830   0.0830 .
## height       -0.2659     0.6775  -0.392   0.6991
## rv            0.3145     0.1648   1.909   0.0715 .
## frc          -0.5492     0.3231  -1.700   0.1055
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.89 on 19 degrees of freedom
## Multiple R-squared:  0.5254, Adjusted R-squared:  0.4005
## F-statistic: 4.207 on 5 and 19 DF,  p-value: 0.00963
```

a. For $\hat{\beta}_{height}$: standard error increased two-fold (from 0.33 to 0.67) - this happened due to collinearity between *height* and *weight*.

```
cor(cystfibr$weight, cystfibr$height)
```

```
## [1] 0.9206953
```

```
R2.height.full <- summary(lm(height ~ sex + weight + rv + frc, data=cystfibr))$r.squared
R2.height.reduced <- summary(lm(height ~ sex + frc, data=cystfibr))$r.squared
R2.height.full
```

```
## [1] 0.8683712
```

```
R2.height.reduced
```

```
## [1] 0.3926207
```

In particular, the variance inflation factor $VIF(\hat{\beta}_{height})$ is much higher in full model (7.597 vs 1.646) due to *height* being explained by the rest of predictors very well (especially by *weight*), hence high R^2_{height} $(= 0.86)$.

From a more intuitive point of view (like explanation in part $1(d)$), due to high collinearity between *height* and *weight*, it is much tougher to accurately estimate effect of *height* while **holding** *weight* **fixed**, as they pretty much **always move together**. Hence, in full model, we get the inflated variance of $\hat{\beta}_{height}$, higher standard error of $\hat{\beta}_{height}$.

For $\hat{\beta}_{frc}$: standard error increased two-fold (from 0.16 to 0.32) as well. Answer if fully analogous to the one for *height* and *weight* in previous two paragraphs.

```
cor(cystfibr$rv, cystfibr$frc)
```

```
## [1] 0.9106029
```

```
R2.frc.full <- summary(lm(frc ~ sex + height + weight + rv , data=cystfibr))$r.squared
R2.frc.reduced <- summary(lm(frc ~ sex + weight, data=cystfibr))$r.squared
R2.frc.full
```

```
## [1] 0.8600279
```

```
R2.frc.reduced
```

```
## [1] 0.3855325
```

b. Getting the VIF of the full model:

```
library(car)
vif(lm(pemax ~ sex + weight + height + rv + frc, data=cystfibr))
```

```
##      sex   weight   height       rv      frc
## 1.113463 7.734391 7.597122 7.193580 7.144279
```

First, we drop *weight* as it has the highest VIF, which is also > 5.

```
vif(lm(pemax ~ sex + height + rv + frc, data=cystfibr))
```

```
##      sex   height       rv      frc
## 1.113227 1.646786 6.268064 6.682708
```

Second, we drop frc as it has the highest remaining VIF, which is also > 5.

```
vif(lm(pemax ~ sex + weight + rv, data=cystfibr))
```

```
##      sex   weight       rv
## 1.080387 1.630741 1.696516
```

All of the remaining VIF values are < 5, we don't have multi-collinearity among these variables.

Problem #3

Code below:

```
library(ISLR)

lm.full <- lm(mpg ~ .-name, data=Auto)
lm.reduced <- step(lm.full)
```

```
## Start:  AIC=950.5
## mpg ~ (cylinders + displacement + horsepower + weight + acceleration +
##     year + origin + name) - name
##
##                Df Sum of Sq    RSS     AIC
## - acceleration  1      7.36 4259.6  949.18
## - horsepower    1     16.74 4269.0  950.04
## <none>                      4252.2  950.50
## - cylinders     1     25.79 4278.0  950.87
## - displacement  1     77.61 4329.8  955.59
## - origin        1    291.13 4543.3  974.46
## - weight        1   1091.63 5343.8 1038.08
## - year          1   2402.25 6654.5 1124.06
##
## Step:  AIC=949.18
## mpg ~ cylinders + displacement + horsepower + weight + year +
##     origin
##
##                Df Sum of Sq    RSS     AIC
## <none>                      4259.6  949.18
## - cylinders     1     27.27 4286.8  949.68
## - horsepower    1     53.80 4313.4  952.10
## - displacement  1     73.57 4333.1  953.89
## - origin        1    292.02 4551.6  973.17
## - weight        1   1310.43 5570.0 1052.32
## - year          1   2396.17 6655.7 1122.13
```

```
lm.reduced
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     year + origin, data = Auto)
##
## Coefficients:
##  (Intercept)     cylinders  displacement    horsepower        weight
##   -15.563492     -0.506685      0.019269     -0.023895     -0.006218
##         year        origin
##     0.747516      1.428242
```

1. *step*() function. *acceleration* is the only variable getting dropped.
2. - "Df" is the # of parameters dropped from the model when we drop the respective predictor (e.g. *acceleration* has $Df = 1$ because it only has one slope parameter associated with it).
   - "Sum of Sq" is the increase in RSS (Residual Sum of Squares) as we drop this predictor.
   - "RSS" is the RSS of the model we obtain after dropping this predictor.
   - "AIC" is the criteria that balances the quality of fit with the complexity of the model, allowing us to select the optimal subset of predictors via minimizing this criteria.
3. *Acceleration* ended up being the only variable dropped from the original full model. It couldn't drop any more variables because that would've led to worse AIC values (worse prediction quality/model complexity trade-off) compared to the the model with all variables intact (the "<none>" line of the output).

Problem #4

```
Advertising <- read.csv("~/Downloads/Advertising.csv")
lm.TV <- lm(sales ~ TV, data=Advertising)
lm.TV.radio <- lm(sales ~ TV + radio, data=Advertising)
lm.TV.radio.new <- lm(sales ~ TV + radio + newspaper, data=Advertising)
```

1. RSS decreased every time we added a predictor (2102 → 556.914 → 556.8253). Whenever we add an extra variable, we can at the very worst repeat the least value for the residual sum of squares (RSS) from the previous model (by setting the slope of that extra variable to 0).

```
print(sum(resid(lm.TV)^2))
```

```
## [1] 2102.531
```

```
print(sum(resid(lm.TV.radio)^2))
```

```
## [1] 556.914
```

```
print(sum(resid(lm.TV.radio.new)^2))
```

```
## [1] 556.8253
```

2. Yes, R^2 increased every time we added a predictor ($0.61187 \to 0.89719 \to 0.89721$). Using the explanation from part (a): adding an extra variable typically decreases the $RSS \implies$ decreases $\frac{RSS}{TSS}$ (as $TSS = \sum_i (y_i - \bar{y})^2$ does **not** depend on predictors being added/dropped, staying constant) $\implies$ increases $R^2 = 1 - \frac{RSS}{TSS}$.

```
summary(lm.TV)$r.squared
```

```
## [1] 0.6118751
```

```
summary(lm.TV.radio)$r.squared
```

```
## [1] 0.8971943
```

```
summary(lm.TV.radio.new)$r.squared
```

```
## [1] 0.8972106
```