Problem Set 3

Zimo Shu

Problem 1

a. Merge the two files to create a single data.frame. Keep only records which matched. Print out the dimensions of the merged data.frame.

```
library(haven)  # read_xpt()
# Citation: from R studio Help:
# Description: The SAS transport format is a open format, as is required for submission of th
# Usage: read_xpt(file, col_select = NULL, skip = 0, n_max = Inf, .name_repair = "unique")

library(knitr)
```

```
aux_i <- read_xpt("AUX_I.XPT")
demo_i <- read_xpt("DEMO_I.XPT")
head(aux_i)
```

```
# A tibble: 6 x 73
   SEQN AUAEXSTS AUAEXCMT AUQ010 AUQ020 AUQ020A AUQ020B AUQ020C AUQ020D AUQ020E
  <dbl>    <dbl>    <dbl>  <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1 83732        1       NA      4      1       2       1       2       2       2
2 83733        1       NA      4      2      NA      NA      NA      NA      NA
3 83735        1       NA      4      2      NA      NA      NA      NA      NA
4 83736        1       NA      4      2      NA      NA      NA      NA      NA
5 83741        2       NA      4      2      NA      NA      NA      NA      NA
6 83742        1       NA      4      2      NA      NA      NA      NA      NA
# i 63 more variables: AUQ030 <dbl>, AUQ040 <dbl>, AUQ050 <dbl>,
#   AUXOTSPL <dbl>, AUXLOEXC <dbl>, AUXLOIMC <dbl>, AUXLOCOL <dbl>,
#   AUXLOABN <dbl>, AUDLOABC <dbl>, AUXROTSP <dbl>, AUXROEXC <dbl>,
#   AUXROIMC <dbl>, AUXROCOL <dbl>, AUXROABN <dbl>, AUDROABC <dbl>,
#   AUXTMEPR <dbl>, AUXTPVR <dbl>, AUXTWIDR <dbl>, AUXTCOMR <dbl>,
#   AUXTMEPL <dbl>, AUXTPVL <dbl>, AUXTWIDL <dbl>, AUXTCOML <dbl>,
#   AUAEAR <dbl>, AUAMODE <dbl>, AUAFMANL <dbl>, AUAFMANR <dbl>, ...
```

```r
head(demo_i)
```

```
# A tibble: 6 x 47
   SEQN SDDSRVYR RIDSTATR RIAGENDR RIDAGEYR RIDAGEMN RIDRETH1 RIDRETH3 RIDEXMON
  <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1 83732        9        2        1       62       NA        3        3        1
2 83733        9        2        1       53       NA        3        3        1
3 83734        9        2        1       78       NA        3        3        2
4 83735        9        2        2       56       NA        3        3        2
5 83736        9        2        2       42       NA        4        4        2
6 83737        9        2        2       72       NA        1        1        1
# i 38 more variables: RIDEXAGM <dbl>, DMQMILIZ <dbl>, DMQADFC <dbl>,
#   DMDBORN4 <dbl>, DMDCITZN <dbl>, DMDYRSUS <dbl>, DMDEDUC3 <dbl>,
#   DMDEDUC2 <dbl>, DMDMARTL <dbl>, RIDEXPRG <dbl>, SIALANG <dbl>,
#   SIAPROXY <dbl>, SIAINTRP <dbl>, FIALANG <dbl>, FIAPROXY <dbl>,
#   FIAINTRP <dbl>, MIALANG <dbl>, MIAPROXY <dbl>, MIAINTRP <dbl>,
#   AIALANGA <dbl>, DMDHHSIZ <dbl>, DMDFMSIZ <dbl>, DMDHHSZA <dbl>,
#   DMDHHSZB <dbl>, DMDHHSZE <dbl>, DMDHRGND <dbl>, DMDHRAGE <dbl>, ...
```

We could see the common column is SEQN.

```r
merged_df <- merge(demo_i, aux_i, by = "SEQN", all = FALSE)
cat("Dimensions:\n")
```

```
Dimensions:
```

```r
print(dim(merged_df))
```

```
[1] 4582  119
```

b. Clean up data.

```r
# First check if there are any weird values
table(merged_df$RIAGENDR, useNA = "ifany")
```

```

   1    2 
2176 2406 
```

```r
table(merged_df$DMDCITZN, useNA = "ifany")
```

```
   1    2    7    9 <NA> 
3684  884    8    5    1 
```

```r
table(merged_df$DMDHHSZA, useNA = "ifany")
```

```
   0    1    2    3 
3463  737  317   65 
```

```r
table(merged_df$INDHHIN2, useNA = "ifany")
```

```
   1    2    3    4    5    6    7    8    9   10   12   13   14   15   77   99 
 101  146  233  262  272  461  439  381  295  244  145   53  458  795   86   63 
<NA> 
 148 
```

```r
# Start cleaning data
# 1. Gender
merged_df$RIAGENDR <- factor(merged_df$RIAGENDR, levels = c(1, 2), labels = c("Male", "Femal


# 2. Citizenship
# 7/9 -> NA
merged_df$DMDCITZN[merged_df$DMDCITZN %in% c(7, 9)] <- NA

merged_df$DMDCITZN <- factor(merged_df$DMDCITZN, levels = c(1, 2), labels = c("Citizen", "No

# 3. Number of children 5 years or younger in the household
merged_df$DMDHHSZA <- as.numeric(merged_df$DMDHHSZA)

# 4. Annual household income
# Replace 77 and 99 with NA
merged_df$INDHHIN2[merged_df$INDHHIN2 %in% c(77, 99)] <- NA
# Issue: 12 = "$20,000 and Over" and 13 = "Under $20,000" are strange orderings
# To fix, I will treat 12 and 13 as NA and reorder
```

```r
merged_df$INDHHIN2[merged_df$INDHHIN2 %in% c(12, 13)] <- NA
income_raw <- merged_df$INDHHIN2
income_levels  <- c(1,2,3,4,5,6,7,8,9,10,14,15)
income_labels  <- c("$0-4,999",
                    "$5,000-9,999",
                    "$10,000-14,999",
                    "$15,000-19,999",
                    "$20,000-24,999",
                    "$25,000-34,999",
                    "$35,000-44,999",
                    "$45,000-54,999",
                    "$55,000-64,999",
                    "$65,000-74,999",
                    "$75,000-99,999",
                    "$100,000 and over")
merged_df$INDHHIN2_clean <- factor(income_raw, levels = income_levels, labels = income_label
```

```r
# Check the cleaned data
cat("Gender:\n")
```

```
Gender:
```

```r
print(table(merged_df$RIAGENDR, useNA="ifany"))
```

```
  Male Female
  2176   2406
```

```r
cat("\nCitizenship:\n")
```

```
Citizenship:
```

```r
print(table(merged_df$DMDCITZN, useNA="ifany"))
```

```
    Citizen Not citizen        <NA>
       3684         884          14
```

```r
cat("\nNumber of children 5 years or younger in the household:\n")
```

```
Number of children 5 years or younger in the household:
```

```r
print(table(merged_df$DMDHHSZA))
```

```
   0    1    2    3 
3463  737  317   65 
```

```r
cat("\nAnnual household income:\n")
```

```
Annual household income:
```

```r
print(table(merged_df$INDHHIN2_clean, useNA="ifany"))
```

```
        $0-4,999      $5,000-9,999    $10,000-14,999    $15,000-19,999 
             101               146               233               262 
  $20,000-24,999    $25,000-34,999    $35,000-44,999    $45,000-54,999 
             272               461               439               381 
  $55,000-64,999    $65,000-74,999    $75,000-99,999 $100,000 and over 
             295               244               458               795 
            <NA> 
             495 
```

c.

```r
# Fit Poisson regression models

# Right ear: gender
m1R <- glm(AUXTWIDR ~ RIAGENDR, data = merged_df,
           family = poisson(link = "log"))

# Right ear: gender + citizenship + children + income
m2R <- glm(AUXTWIDR ~ RIAGENDR + DMDCITZN +
             DMDHHSZA + INDHHIN2_clean,
```

```r
           data = merged_df, family = poisson(link = "log"))

# Left ear: gender
m1L <- glm(AUXTWIDL ~ RIAGENDR, data = merged_df,
           family = poisson(link = "log"))

# Left ear: gender + citizenship + children + income
m2L <- glm(AUXTWIDL ~ RIAGENDR + DMDCITZN +
             DMDHHSZA + INDHHIN2_clean,
           data = merged_df, family = poisson(link = "log"))
```

```r
m1R
```

```
Call:  glm(formula = AUXTWIDR ~ RIAGENDR, family = poisson(link = "log"),
    data = merged_df)

Coefficients:
   (Intercept)  RIAGENDRFemale
      4.434588        0.009264

Degrees of Freedom: 4148 Total (i.e. Null);  4147 Residual
  (433 observations deleted due to missingness)
Null Deviance:      70970
Residual Deviance: 70960    AIC: 96620
```

```r
m2R
```

```
Call:  glm(formula = AUXTWIDR ~ RIAGENDR + DMDCITZN + DMDHHSZA + INDHHIN2_clean,
    family = poisson(link = "log"), data = merged_df)

Coefficients:
        (Intercept)       RIAGENDRFemale  DMDCITZNNot citizen
          4.4256110            0.0151278            0.0412489
           DMDHHSZA     INDHHIN2_clean.L     INDHHIN2_clean.Q
         -0.0040778           -0.0493448           -0.0807019
   INDHHIN2_clean.C     INDHHIN2_clean^4     INDHHIN2_clean^5
         -0.0001513            0.0226155           -0.0047230
   INDHHIN2_clean^6     INDHHIN2_clean^7     INDHHIN2_clean^8
          0.0523611           -0.0032544            0.0294443
```

```
   INDHHIN2_clean^9    INDHHIN2_clean^10    INDHHIN2_clean^11
          0.0342785           -0.0207319            0.0044830

Degrees of Freedom: 3704 Total (i.e. Null);  3690 Residual
  (877 observations deleted due to missingness)
Null Deviance:      62370
Residual Deviance: 61750    AIC: 84680
```

```
m1L
```

```
Call:  glm(formula = AUXTWIDL ~ RIAGENDR, family = poisson(link = "log"),
    data = merged_df)

Coefficients:
   (Intercept)  RIAGENDRFemale
       4.43981         0.01295

Degrees of Freedom: 4102 Total (i.e. Null);  4101 Residual
  (479 observations deleted due to missingness)
Null Deviance:      73310
Residual Deviance: 73290    AIC: 98690
```

```
m2L
```

```
Call:  glm(formula = AUXTWIDL ~ RIAGENDR + DMDCITZN + DMDHHSZA + INDHHIN2_clean,
    family = poisson(link = "log"), data = merged_df)

Coefficients:
        (Intercept)       RIAGENDRFemale  DMDCITZNNot citizen
           4.444469             0.018747             0.018580
           DMDHHSZA     INDHHIN2_clean.L     INDHHIN2_clean.Q
          -0.017432            -0.070060            -0.016554
   INDHHIN2_clean.C     INDHHIN2_clean^4     INDHHIN2_clean^5
           0.024307            -0.024130            -0.007941
   INDHHIN2_clean^6     INDHHIN2_clean^7     INDHHIN2_clean^8
           0.060115            -0.023379            -0.006481
   INDHHIN2_clean^9    INDHHIN2_clean^10    INDHHIN2_clean^11
           0.006410            -0.031142            -0.050659
```

```
Degrees of Freedom: 3664 Total (i.e. Null);  3650 Residual
  (917 observations deleted due to missingness)
Null Deviance:      64150
Residual Deviance: 63700    AIC: 86400
```

Based on the outputs, we created a nice table. (I wrote a summary function to do this at first, but I encountered an unexpected c stack usage limit error, which I failed to solve. So I changed my way using a direct computation of statistics but this could look a little complicated. I would definitely ask about the c stack error during OH or after class next week.)

```r
# AIC
AIC_1R <- 84680
AIC_1L <- 98690
AIC_2R <- 86400
AIC_2L <- 96620

# n
n_1R <- 3704 + 1   # 3705
n_1L <- 4102 + 1   # 4103
n_2R <- 3664 + 1   # 3665
n_2L <- 4148 + 1   # 4149

# pseudo-R^2 = 1 - Residual / Null
null_1R <- 62370; res_1R <- 61750
null_1L <- 73310; res_1L <- 73290
null_2R <- 64150; res_2R <- 63700
null_2L <- 70970; res_2L <- 70960

pseudo_1R <- 1 - res_1R / null_1R
pseudo_1L <- 1 - res_1L / null_1L
pseudo_2R <- 1 - res_2R / null_2R
pseudo_2L <- 1 - res_2L / null_2L

# Coefficient estimates
# Gender
b_1R_gender <- 0.009264
b_1L_gender <- 0.01295
b_2R_gender <- 0.0151278
b_2L_gender <- 0.018747

# Citizenship
b_2R_cit <- 0.0412489
```

```r
b_2L_cit <- 0.01858

# Children
b_2R_kids <- -0.0040778
b_2L_kids <- -0.017432

irr_1R_gender <- exp(b_1R_gender)
irr_1L_gender <- exp(b_1L_gender)
irr_2R_gender <- exp(b_2R_gender)
irr_2L_gender <- exp(b_2L_gender)

irr_2R_cit    <- exp(b_2R_cit)
irr_2L_cit    <- exp(b_2L_cit)

irr_2R_kids   <- exp(b_2R_kids)
irr_2L_kids   <- exp(b_2L_kids)

coef_tab <- data.frame(
  Model = c("1R","2R","1L","2L"),
  Ear   = c("Right","Right","Left","Left"),
  `Gender (F vs M)` = round(c(irr_1R_gender, irr_2R_gender, irr_1L_gender, irr_2L_gender), 3
  `Citizenship (Not vs Cit)` = round(c(NA, irr_2R_cit, NA, irr_2L_cit), 3),
  `Children (per child)` = round(c(NA, irr_2R_kids, NA, irr_2L_kids), 3),
  check.names = FALSE
)

# replace NA with em dash
coef_tab[is.na(coef_tab)] <- "-"

# Model stats table
stats_tab <- data.frame(
  Model = c("1R","2R","1L","2L"),
  Ear   = c("Right","Right","Left","Left"),
  n     = c(n_1R, n_2R, n_1L, n_2L),
  `Pseudo-R^2` = round(c(pseudo_1R, pseudo_2R, pseudo_1L, pseudo_2L), 3),
  AIC   = c(AIC_1R, AIC_2R, AIC_1L, AIC_2L),
  check.names = FALSE
)
```

```r
knitr::kable(coef_tab,
             caption = "IRR",
             align = c("l","l","r","r","r"))
```

Table 1: IRR

Model	Ear	Gender (F vs M)	Citizenship (Not vs Cit)	Children (per child)
1R	Right	1.009	—	—
2R	Right	1.015	1.042	0.996
1L	Left	1.013	—	—
2L	Left	1.019	1.019	0.983

```r
knitr::kable(stats_tab,
             caption = "Model Summary Statistics",
             align = c("l","l","r","r","r"),
             digits = 3)
```

Table 2: Model Summary Statistics

Model	Ear	n	Pseudo-R^2	AIC
1R	Right	3705	0.010	84680
2R	Right	3665	0.007	86400
1L	Left	4103	0.000	98690
2L	Left	4149	0.000	96620

d. From model 2L, provide evidence whether there is a difference between males and females in terms of their incidence risk ratio. Test whether the predicted value of Tympanometric width measure of the left ear differs between men and women. Include the results of the each test and their interpretation.

```r
# Wald test on the gender coefficient
b  <- coef(m2L)["RIAGENDRFemale"]
se <- sqrt(vcov(m2L)["RIAGENDRFemale","RIAGENDRFemale"])
z  <- b / se
p  <- 2 * (1 - pnorm(abs(z)))

IRR <- exp(b)
CI  <- exp(b + c(-1, 1) * 1.96 * se)

cat(sprintf("Model 2L Female vs Male:\n"))
```

```
Model 2L Female vs Male:
```

```
cat(sprintf(" IRR = %.3f, 95%% CI [%.3f, %.3f], z = %.2f, p = %.3g\n\n",
            IRR, CI[1], CI[2], z, p))
```

```
 IRR = 1.019, 95% CI [1.012, 1.026], z = 5.20, p = 1.95e-07
```

```
# Test 2: predict
# Citation: from Rstudio help: levels provides access to the levels attribute of a variable.

lev_g   <- levels(merged_df$RIAGENDR)
lev_cit <- levels(merged_df$DMDCITZN)
lev_inc <- levels(merged_df$INDHHIN2_clean)

mode_level <- function(x) names(which.max(table(x)))

cit_typ <- mode_level(merged_df$DMDCITZN)
inc_typ <- mode_level(merged_df$INDHHIN2_clean)
kids_mu <- mean(merged_df$DMDHHSZA, na.rm = TRUE)

newdata <- data.frame(
  RIAGENDR  = factor(c("Male","Female"), levels = lev_g),
  DMDCITZN  = factor(rep(cit_typ, 2),        levels = lev_cit),
  DMDHHSZA = kids_mu,
  INDHHIN2_clean = factor(rep(inc_typ, 2),        levels = lev_inc, ordered = TRUE)
)

# Predicted means
pred_mu <- predict(m2L, newdata, type = "response")

X <- model.matrix(m2L, newdata)       # design matrix
V <- vcov(m2L)                        # covariance
cvec <- X[2, ] - X[1, ]
d_eta <- drop(cvec %*% coef(m2L))     # difference
se_d <- sqrt(drop(t(cvec) %*% V %*% cvec))
z_d <- d_eta / se_d
p_d <- 2 * (1 - pnorm(abs(z_d)))
# ratio of predicted means
IRR_pred <- exp(d_eta)

cat("Predicted means at typical covariates:\n")
```

```
Predicted means at typical covariates:
```

```r
print(data.frame(Gender = c("Male","Female"),
                 Predicted_LeftEar = round(pred_mu, 2)),
      row.names = FALSE)
```

```
 Gender Predicted_LeftEar
   Male             81.90
 Female             83.45
```

```r
cat(sprintf("\nTest of difference in predicted means (Female - Male):\n"))
```

```
Test of difference in predicted means (Female - Male):
```

```r
cat(sprintf("z = %.2f, p = %.3g\n", z_d, p_d))
```

```
z = 10.03, p = 0
```

```r
cat(sprintf("Ratio of predicted means (Female/Male) = %.3f\n", IRR_pred))
```

```
Ratio of predicted means (Female/Male) = 1.116
```

Wald test:

We could see that from Model 2L, the estimated incidence rate ratio for females relative to males is 1.019. So holding citizenship, number of children, and household income unchanged, the expected tympanometric width in the left ear is approximately 1.9% higher for females than for males. The p-valu is pretty small, which indicates that this difference is statistically significant.

Predicted-value test:

83.45 vs 81.90; small p value; females also show a higher expected tympanometric width than males.

In conclusion, there is a statistically significant but actually modest difference.

Problem 2 - Sakila

```r
library(DBI)
```

```
Warning: package 'DBI' was built under R version 4.3.3
```

```r
library(RSQLite)
```

```
Warning: package 'RSQLite' was built under R version 4.3.3
```

```r
sakila <- dbConnect(RSQLite::SQLite(), "sakila_master.db")
dbListTables(sakila)
```

```
 [1] "actor"                  "address"                "category"
 [4] "city"                   "country"                "customer"
 [7] "customer_list"          "film"                   "film_actor"
[10] "film_category"          "film_list"              "film_text"
[13] "inventory"              "language"               "payment"
[16] "rental"                 "sales_by_film_category" "sales_by_store"
[19] "staff"                  "staff_list"             "store"
```

a. For each store, how many customers does that store have, and what percentage of customers of that store are active in the system?

```r
# Get the list of all columns for customer table
dbListFields(sakila, "customer")
```

```
[1] "customer_id" "store_id"    "first_name"  "last_name"   "email"
[6] "address_id"  "active"      "create_date" "last_update"
```

```r
dbGetQuery(sakila, "SELECT * FROM customer LIMIT 5")
```

```
  customer_id store_id first_name last_name                               email
1           1        1       MARY     SMITH       MARY.SMITH@sakilacustomer.org
2           2        1   PATRICIA   JOHNSON PATRICIA.JOHNSON@sakilacustomer.org
3           3        1      LINDA  WILLIAMS   LINDA.WILLIAMS@sakilacustomer.org
4           4        2    BARBARA     JONES    BARBARA.JONES@sakilacustomer.org
5           5        1  ELIZABETH     BROWN  ELIZABETH.BROWN@sakilacustomer.org
  address_id active             create_date         last_update
1          5      1 2006-02-14 22:04:36.000 2020-12-23 07:15:11
```

```
2          6      1 2006-02-14 22:04:36.000 2020-12-23 07:15:11
3          7      1 2006-02-14 22:04:36.000 2020-12-23 07:15:11
4          8      1 2006-02-14 22:04:36.000 2020-12-23 07:15:11
5          9      1 2006-02-14 22:04:36.000 2020-12-23 07:15:11
```

```
# Take a look at the active variable
dbGetQuery(sakila, "
SELECT store_id, active
  FROM customer
 LIMIT 5
")
```

```
  store_id active
1        1      1
2        1      1
3        1      1
4        2      1
5        1      1
```

```
dbGetQuery(sakila, "
SELECT
  store_id,
  COUNT(*) AS num_customers,
  ROUND(AVG(active) * 100.0, 2) AS percent_active
FROM customer
GROUP BY store_id
ORDER BY store_id;
")
```

```
  store_id num_customers percent_active
1        1           326          97.55
2        2           273          97.44
```

Hence, for store 1, it has 326 customers, and the active customers percentage is 97.55 %. For store 2, it has 273 customers, and the active customers percentage is 97.44 %.

b. Generate a table identifying the names and country of each staff member.

```
# Take a look at staff table first
dbListFields(sakila, "staff")
```

```
 [1] "staff_id"    "first_name"  "last_name"   "address_id"  "picture"
 [6] "email"       "store_id"    "active"      "username"    "password"
[11] "last_update"
```

```r
dbGetQuery(sakila, "SELECT staff_id, first_name, last_name, address_id FROM staff")
```

```
  staff_id first_name last_name address_id
1        1       Mike   Hillyer          3
2        2        Jon  Stephens          4
```

```r
# The staff table links with address, city and country
dbListFields(sakila, "address")
```

```
[1] "address_id"  "address"     "address2"    "district"    "city_id"
[6] "postal_code" "phone"       "last_update"
```

```r
dbGetQuery(sakila, "SELECT address_id, city_id FROM address LIMIT 5")
```

```
  address_id city_id
1          1     300
2          2     576
3          3     300
4          4     576
5          5     463
```

```r
dbListFields(sakila, "city")
```

```
[1] "city_id"     "city"        "country_id"  "last_update"
```

```r
dbGetQuery(sakila, "SELECT city_id, country_id FROM city LIMIT 5")
```

```
  city_id country_id
1       1         87
2       2         82
3       3        101
4       4         60
5       5         97
```

```r
dbListFields(sakila, "country")
```

```
[1] "country_id"  "country"     "last_update"
```

```r
dbGetQuery(sakila, "SELECT country_id, country FROM country LIMIT 5")
```

```
  country_id        country
1          1    Afghanistan
2          2        Algeria
3          3 American Samoa
4          4         Angola
5          5       Anguilla
```

```r
dbGetQuery(sakila, "
SELECT
  s.staff_id,
  s.first_name || ' ' || s.last_name AS staff_name,
  co.country
FROM staff   AS s
JOIN address AS a  ON a.address_id = s.address_id
JOIN city    AS ci ON ci.city_id = a.city_id
JOIN country AS co ON co.country_id = ci.country_id
ORDER BY s.staff_id;
")
```

```
  staff_id   staff_name   country
1        1 Mike Hillyer    Canada
2        2 Jon Stephens Australia
```

c. Identify the name(s) of the film(s) which was/were rented for the highest dollar value. (Assume all costs are in USD regardless of country.) (Hint: You can merge a table more than once.)

```r
# Find the one with highest value here
dbGetQuery(sakila, "
WITH values_film AS (
  SELECT
      f.film_id,
      f.title,
      SUM(p.amount) AS total_val
```

```
  FROM payment   AS p
  JOIN rental    AS r  ON r.rental_id = p.rental_id
  JOIN inventory AS i  ON i.inventory_id = r.inventory_id
  JOIN film      AS f  ON f.film_id = i.film_id
  GROUP BY f.film_id, f.title
)
SELECT film_id, title, ROUND(total_val, 2) AS total_val
FROM values_film
WHERE total_val = (SELECT MAX(total_val) FROM values_film)
ORDER BY title;
")
```

```
  film_id            title total_val
1     879 TELEGRAPH VOYAGE    231.73
```

Hence, the film that rented for the highest value is TELEGRAPH VOYAGE (with 231.73 dollar value).

Problem 3

```
au <- read.csv("au-500.csv", stringsAsFactors = FALSE)
head(au)
```

```
  first_name last_name                   company_name              address
1   Rebbecca     Didio         Brandt, Jonathan F Esq         171 E 24th St
2     Stevie     Hallo     Landrum Temporary Services        22222 Acoma St
3     Mariko    Stayer            Inabinet, Macre Esq 534 Schoenborn St #51
4    Gerardo    Woodka       Morris Downing & Sherred     69206 Jackson Ave
5      Mayra      Bena            Buelt, David L Esq     808 Glen Cove Ave
6     Idella  Scotland Artesian Ice & Cold Storage Co      373 Lafayette St
        city state post       phone1       phone2                         email
1      Leith   TAS 7315 03-8174-9123 0458-665-290 rebbecca.didio@didio.com.au
2    Proston   QLD 4613 07-9997-3366 0497-622-620    stevie.hallo@hotmail.com
3      Hamel    WA 6215 08-5558-9019 0427-885-282   mariko_stayer@hotmail.com
4   Talmalmo   NSW 2640 02-6044-4682 0443-795-912  gerardo_woodka@hotmail.com
5  Lane Cove   NSW 1595 02-1455-6085 0453-666-885        mayra.bena@gmail.com
6 Cartmeticup    WA 6316 08-7868-1355 0451-966-921          idella@hotmail.com
                                        web
1      http://www.brandtjonathanfesq.com.au
```

```
2 http://www.landrumtemporaryservices.com.au
3         http://www.inabinetmacreesq.com.au
4     http://www.morrisdowningsherred.com.au
5           http://www.bueltdavidlesq.com.au
6 http://www.artesianicecoldstorageco.com.au
```

a. What percentage of the websites are .com's (as opposed to .net, .com.au, etc)?

```r
# Based on R studio Help: trimws Remove leading and/or trailing whitespace from character st
website <- tolower(trimws(au$web))

# Remove scheme and www
website <- sub("^https?://", "", website)
website <- sub("^www\\.", "", website)

## Keep the host
host <- sub("/.*$", "", website)

valid_web <- !is.na(host) & nzchar(host)
# Ends with '.com' exactly -> not .com.au
is_com <- grepl("\\.com$", host)

percentage <- 100 * sum(is_com[valid_web]) / sum(valid_web)
cat("Percentage of .com websites:", percentage, "%\n")
```

```
Percentage of .com websites: 0 %
```

Hence the percentage is 0 % (no ".com" websites in the data).

b. What is the most common domain name amongst the email addresses?

```r
emails <- au$email
domains <- sub(".*@", "", emails)          # after '@'
valid <- !is.na(domains) & nzchar(domains)

table_email <- sort(table(domains[valid]), decreasing = TRUE)
common_domain <- names(table_email)[1]
cat("Most common email domain:", common_domain, "\n")
```

```
Most common email domain: hotmail.com
```

Hence, the most common email domain name is hotmail.com.

c. What proportion of company names contain a non-alphabetic character, excluding commas and whitespace. (E.g. "Jane Doe, LLC" would not contain an eligible non-alphabetic character; "Plumber 247" would.) What about if you also exclude ampersands ("&")?

```
companies <- au$company_name

# Excluding commas and whitespace
clean_company <- gsub("[,[:space:]]", "", companies)

nonalpha <- grepl("[^A-Za-z]", clean_company)
prop_nonalpha <- mean(nonalpha, na.rm = TRUE)

# Excluding ampersands
clean_company2 <- gsub("&", "", clean_company)
nonalpha_noamp <- grepl("[^A-Za-z]", clean_company2)
prop_nonalpha_noamp <- mean(nonalpha_noamp, na.rm = TRUE)

cat("Proportion containing non-alphabetic character (excluding commas & spaces): ", 100*prop_
```

```
Proportion containing non-alphabetic character (excluding commas & spaces): 9%
```

```
cat("Proportion containing non-alphabetic character (excluding '&' as well): ", 100*prop_non
```

```
Proportion containing non-alphabetic character (excluding '&' as well): 0.8%
```

Hence, we could see that proportion containing non-alphabetic character (excluding commas & spaces) is about 9%. If we excluded '&' as well, the proportion dropped to only 0.8%.

d. Cell phones: 1234-567-890 There are two different phones listed for each record. Make all phone numbers written like cell phones. Show it works by printing the first 10 phone numbers of each column.

```
# Function to format the phone number like cell phone
cell_phone <- function(x) {
  x <- gsub("\\D", "", x)  # keep digits
  sub("^(\\d{4})(\\d{3})(\\d{3})$", "\\1-\\2-\\3", x)
}
```

```r
au$phone1_cell <- cell_phone(au$phone1)
au$phone2_cell <- cell_phone(au$phone2)

head(cbind(cell1 = au$phone1_cell, cell2 = au$phone2_cell), 10)
```

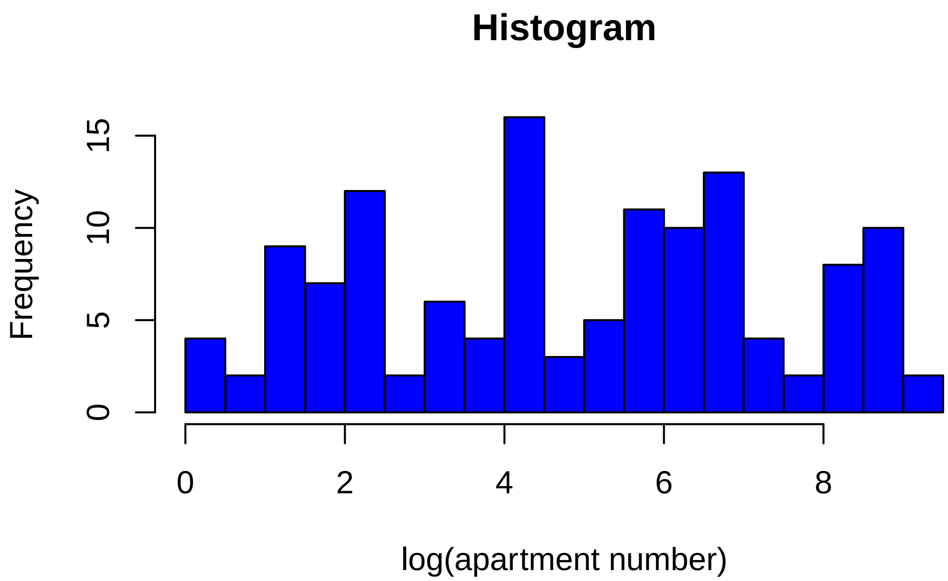
```
      cell1          cell2
 [1,] "0381-749-123" "0458-665-290"
 [2,] "0799-973-366" "0497-622-620"
 [3,] "0855-589-019" "0427-885-282"
 [4,] "0260-444-682" "0443-795-912"
 [5,] "0214-556-085" "0453-666-885"
 [6,] "0878-681-355" "0451-966-921"
 [7,] "0865-228-931" "0427-991-688"
 [8,] "0252-269-402" "0415-961-606"
 [9,] "0731-849-989" "0411-732-965"
[10,] "0868-904-661" "0461-862-457"
```

e. Produce a histogram of the log of the apartment numbers for all addresses. (You may assume any number at the end of the an address is an apartment number.)

```r
addr <- au$address

apt_str <- sub(".*?(\\d+)$", "\\1", addr)
# Convert to numeric
apt <- as.numeric(apt_str)
```

```
Warning: NAs introduced by coercion
```

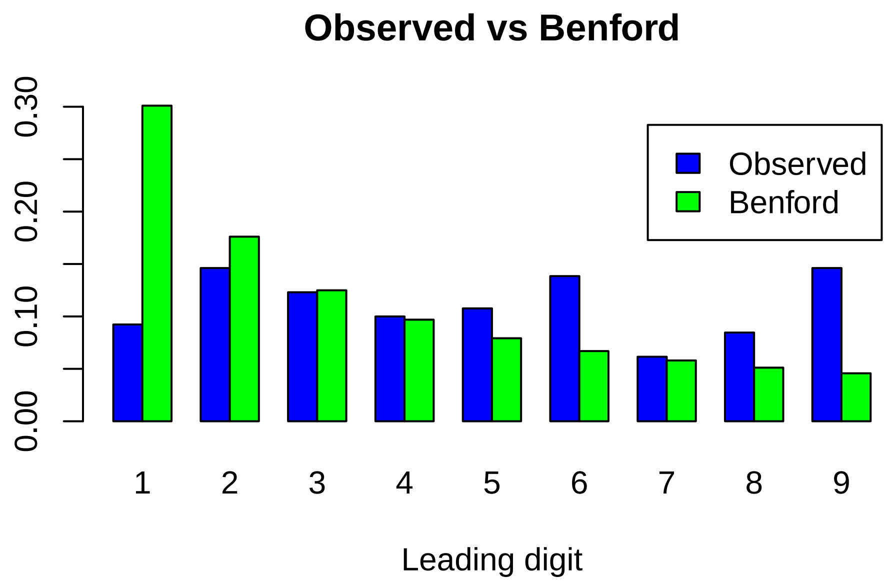
```r
apt <- apt[!is.na(apt) & apt > 0]

hist(log(apt), breaks = 20,
     main = "Histogram",
     xlab = "log(apartment number)",
     col = "blue", border = "black")
```



f. Benford's law is an observation about the distribution of the leading digit of real numerical data. Examine whether the apartment numbers appear to follow Benford's law. Do you think the apartment numbers would pass as real data?

```
# Drop any 0
leading_digit <- as.numeric(substr(as.character(apt), 1, 1))
leading_digit <- leading_digit[leading_digit %in% 1:9]

observe <- table(leading_digit)
observe_prop <- observe / sum(observe)

# Benford expected probabilities
# citation: according to wiki page: https://en.wikipedia.org/wiki/Benford%27s_law
# In sets that obey the law, the number 1 appears as the leading significant digit about 30%
# P(d) = log10(d+1) - log10(d)

benford <- log10(1 + 1/(1:9))
names(benford) <- as.character(1:9)

barplot(rbind(observe_prop, benford),
        beside = TRUE, col = c("blue", "green"),
        legend = c("Observed", "Benford"),
```

```
        main = "Observed vs Benford",
        xlab = "Leading digit")
```




```
cbind(Observed = observe_prop, Benford = benford)
```

```
    Observed    Benford
1 0.09230769 0.30103000
2 0.14615385 0.17609126
3 0.12307692 0.12493874
4 0.10000000 0.09691001
5 0.10769231 0.07918125
6 0.13846154 0.06694679
7 0.06153846 0.05799195
8 0.08461538 0.05115252
9 0.14615385 0.04575749
```

In observed data, we can see that the pattern is pretty different. Hence, based on my obeservation, I don't think the apartment numbers would pass as real data.