Session 5 Live Coding

Load Libraries

```
library(tidyverse)
library(broom)
library(skimr)
```

Load Data

```
churn.df <- read_csv("churn.csv")
churn.df
```

```
## # A tibble: 7,043 x 19
##    Gender SeniorCitizen Dependents MonthsTenure PhoneService MultipleLines
##    <chr>          <dbl> <chr>             <dbl> <chr>        <chr>
##  1 Female             0 No                    1 No           No phone ser~
##  2 Male               0 No                   34 Yes          No
##  3 Male               0 No                    2 Yes          No
##  4 Male               0 No                   45 No           No phone ser~
##  5 Female             0 No                    2 Yes          No
##  6 Female             0 No                    8 Yes          Yes
##  7 Male               0 Yes                  22 Yes          Yes
##  8 Female             0 No                   10 No           No phone ser~
##  9 Female             0 No                   28 Yes          Yes
## 10 Male               0 Yes                  62 Yes          No
## # ... with 7,033 more rows, and 13 more variables: InternetService <chr>,
## #   OnlineSecurity <chr>, OnlineBackup <chr>, DeviceProtection <chr>,
## #   SupportContract <chr>, StreamingTV <chr>, StreamingMovies <chr>,
## #   Contract <chr>, PaperlessBilling <chr>, PaymentMethod <chr>,
## #   MonthlyRevenue <dbl>, LifetimeRevenue <dbl>, Churn <dbl>
```

Notes

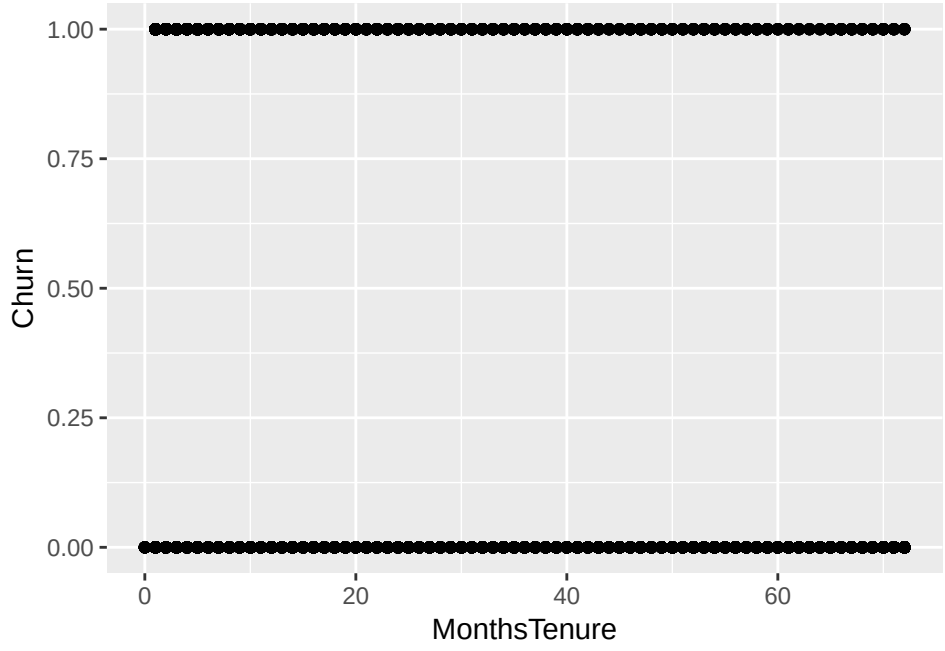
- Churn = 1 when a customer has cancelled his or her contract
- SeniorCitizen = 1 when the customer is 65+

Data Wrangling

```
churn.df <- churn.df %>%
  mutate_if(is.character, as_factor)
```

Why we need to think about LDV models differently...

```
ggplot(data = churn.df, aes(y = Churn, x = MonthsTenure)) +
  geom_point()
```



Summaries

```
# General churn
churn.df %>%
  count(Churn) %>%
  mutate(Ratio = round(n / sum(n), 2))
```

```
## # A tibble: 2 x 3
##   Churn     n Ratio
##   <dbl> <int> <dbl>
## 1     0  5174  0.73
## 2     1  1869  0.27
```

```
# Churn by age
churn.df %>%
  count(Churn, SeniorCitizen) %>%
  mutate(Ratio = round(n / sum(n), 2))
```

```
## # A tibble: 4 x 4
##   Churn SeniorCitizen     n Ratio
##   <dbl>         <dbl> <int> <dbl>
## 1     0             0  4508  0.64
## 2     0             1   666  0.09
## 3     1             0  1393  0.2
## 4     1             1   476  0.07
```

```
# Churn by age and gender
churn.df %>%
  count(Churn, SeniorCitizen, Contract) %>%
  mutate(Ratio = round(n / sum(n), 2))
```

```
## # A tibble: 12 x 5
```

```
##    Churn SeniorCitizen Contract           n Ratio
##    <dbl>         <dbl> <fct>          <int> <dbl>
##  1     0             0 Month-to-month  1854  0.26
##  2     0             0 One year        1146  0.16
##  3     0             0 Two year        1508  0.21
##  4     0             1 Month-to-month   366  0.05
##  5     0             1 One year         161  0.02
##  6     0             1 Two year         139  0.02
##  7     1             0 Month-to-month  1214  0.17
##  8     1             0 One year         137  0.02
##  9     1             0 Two year          42  0.01
## 10     1             1 Month-to-month   441  0.06
## 11     1             1 One year          29  0
## 12     1             1 Two year           6  0
```

```
# Deeper into the churn
churn.df %>%
  filter(Churn == 1) %>%
  count(SeniorCitizen, Contract)
```

```
## # A tibble: 6 x 3
##   SeniorCitizen Contract           n
##           <dbl> <fct>          <int>
## 1             0 Month-to-month  1214
## 2             0 One year         137
## 3             0 Two year          42
## 4             1 Month-to-month   441
## 5             1 One year          29
## 6             1 Two year           6
```

Customer value of churn

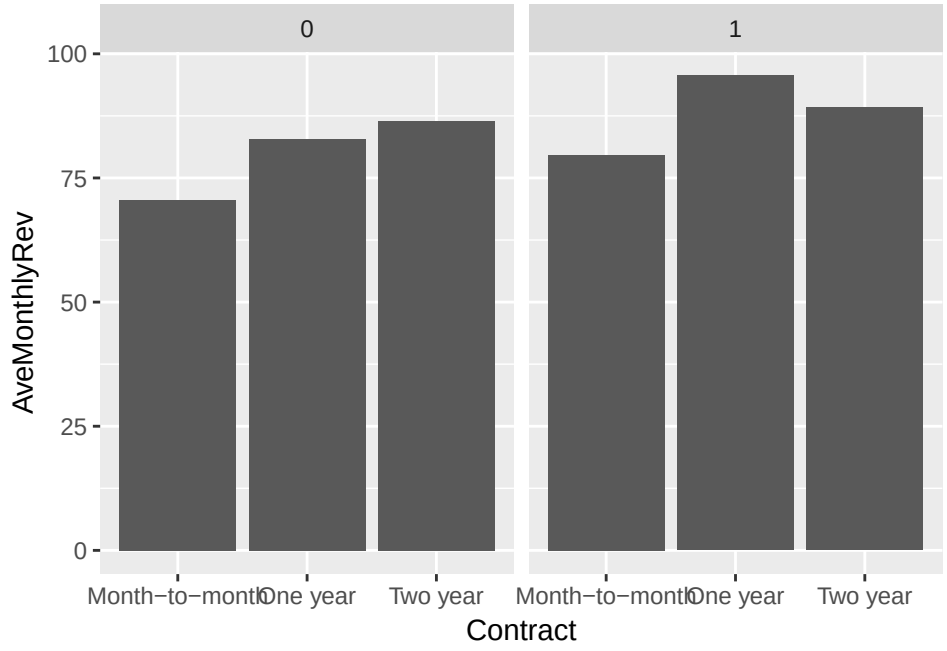
```
# Getting some average values
churn.df %>%
  filter(Churn == 1) %>%
  group_by(SeniorCitizen, Contract) %>%
    summarise(Count = n(),
              AveMonthlyRev = mean(MonthlyRevenue),
              AveLifetimeRev = mean(LifetimeRevenue))
```

```
## # A tibble: 6 x 5
## # Groups:   SeniorCitizen [2]
##   SeniorCitizen Contract       Count AveMonthlyRev AveLifetimeRev
##           <dbl> <fct>          <int>         <dbl>          <dbl>
## 1             0 Month-to-month  1214          70.6          1013.
## 2             0 One year         137          82.8          3846.
## 3             0 Two year          42          86.4          5343.
## 4             1 Month-to-month   441          79.6          1583.
## 5             1 One year          29          95.6          5107.
## 6             1 Two year           6          89.2          6058.
```

What does this picture look like?

```
ggplot(data = churn.df  %>%
  filter(Churn == 1) %>%
```

```
  group_by(SeniorCitizen, Contract) %>%
    summarise(AveMonthlyRev = mean(MonthlyRevenue)),
  aes(y = AveMonthlyRev, x = Contract)) +
  geom_bar(stat = "identity") +
  facet_wrap(~ SeniorCitizen)
```



Model construction

To estimate our LDV model, we need to use a different approach, and for that, we need **logs**, or the logarithimic transformation.

- Think about things in terms of proportions, rather than absolutes. Why does a difference of one year seem like a big amount when you are 10, but a smaller amount when you are 40?

The probability of a customer leaving...

```
contract.model <- glm(Churn ~ Contract, data = churn.df, family = "binomial")
summary(contract.model)
```

```
##
## Call:
## glm(formula = Churn ~ Contract, family = "binomial", data = churn.df)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.056  -1.056  -0.489   1.304   2.670
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -0.29371    0.03248  -9.044   <2e-16 ***
## ContractOne year -1.76980    0.08857 -19.983   <2e-16 ***
## ContractTwo year -3.24180    0.14998 -21.614   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8150.1  on 7042  degrees of freedom
## Residual deviance: 6763.3  on 7040  degrees of freedom
## AIC: 6769.3
##
## Number of Fisher Scoring iterations: 6
```

What you are looking at are the log-odds—lets do the conversion to odds.

```
tidy(contract.model) %>%
  select(term, estimate) %>%
  mutate(Odds = exp(estimate)) %>%
  mutate_if(is.numeric, funs(round(., 3)))
```

```
## # A tibble: 3 x 3
##   term             estimate  Odds
##   <chr>               <dbl> <dbl>
## 1 (Intercept)        -0.294 0.745
## 2 ContractOne year   -1.77  0.17
## 3 ContractTwo year   -3.24  0.039
```

If you see an odds ratio below 1.0, that means that the odds of what you are predicting to happen actually went down with an increase in your independent variable.

You can never have a negative odds ratio, because you can never have a negative probability.

If the odds ratio was 1.0, you would have an equivalent probability (a 50/50 chance) of the y = 1 condition occurring.

Predicted probabilities

```
contract.df <- churn.df %>%
  select(Contract) %>%
  distinct()

contract.pred <- augment(contract.model, newdata = contract.df, type.predict = "response")
contract.pred
```

```
## # A tibble: 3 x 3
##   Contract       .fitted .se.fit
##   <fct>            <dbl>   <dbl>
## 1 Month-to-month  0.427  0.00795
## 2 One year        0.113  0.00824
## 3 Two year        0.0283 0.00403
```

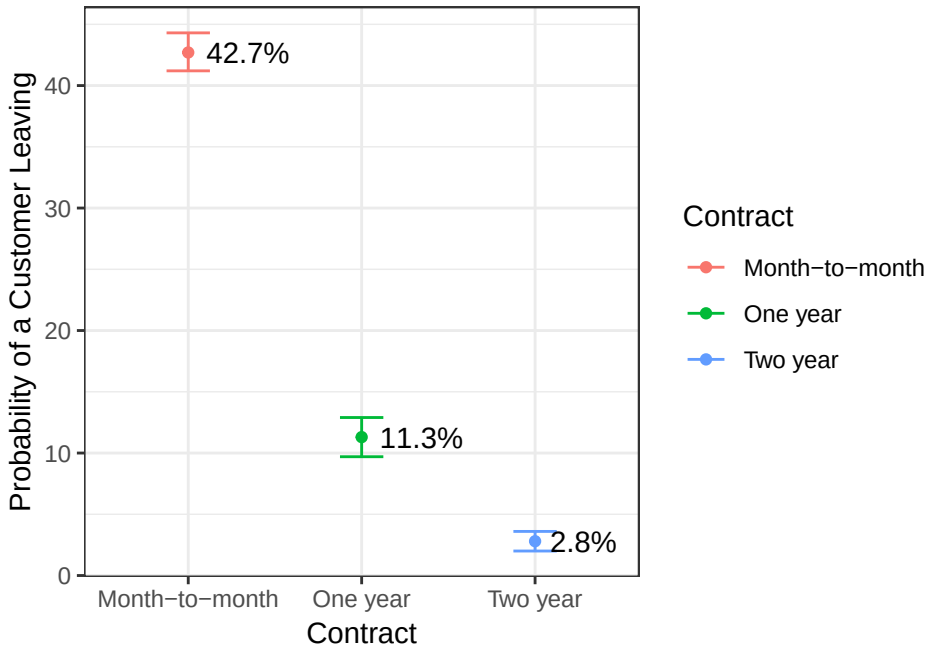
```
# Quantify the uncertainty
contract.pred <- contract.pred %>%
  mutate(Pr_y = 100 * .fitted,
         lower.ci = 100 * (.fitted - (1.96 * .se.fit)),
         upper.ci = 100 * (.fitted + (1.96 * .se.fit))) %>%
  select(Contract, Pr_y, lower.ci, upper.ci) %>%
  mutate_if(is.numeric, funs(round(., 1)))
```

```
contract.pred
```

```
## # A tibble: 3 x 4
##   Contract        Pr_y lower.ci upper.ci
##   <fct>          <dbl>    <dbl>    <dbl>
## 1 Month-to-month  42.7     41.2     44.3
## 2 One year        11.3      9.7     12.9
## 3 Two year         2.8      2        3.6
```

Visualizing predicted probabilities

```
ggplot(data = contract.pred, aes(y = Pr_y, x = Contract, color = Contract)) +
  geom_point() +
  geom_errorbar(aes(ymin = lower.ci, ymax = upper.ci), width = 0.25) +
  geom_text(aes(label = paste0(Pr_y, "% ")), color = "black",
            hjust = - .2) +
  labs(y = "Probability of a Customer Leaving") +
  theme_bw()
```



Multivariate Effects

```
mult.model <- glm(Churn ~ Contract + SeniorCitizen, data = churn.df,
                  family = "binomial")
summary(mult.model)
```

```
##
## Call:
## glm(formula = Churn ~ Contract + SeniorCitizen, family = "binomial",
##     data = churn.df)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2480  -1.0061  -0.4685   1.1085   2.6932
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -0.41741    0.03630 -11.498  < 2e-16 ***
## ContractOne year -1.73685    0.08888 -19.542  < 2e-16 ***
## ContractTwo year -3.18224    0.15022 -21.184  < 2e-16 ***
## SeniorCitizen     0.58177    0.07351   7.914 2.49e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8150.1  on 7042  degrees of freedom
## Residual deviance: 6701.3  on 7039  degrees of freedom
## AIC: 6709.3
##
## Number of Fisher Scoring iterations: 6
```

Predicted probabilities

```
# Create our new dataframe
mult.df <- churn.df %>%
  select(Contract, SeniorCitizen) %>%
  distinct()

mult.df
```

```
## # A tibble: 6 x 2
##   Contract       SeniorCitizen
##   <fct>                  <dbl>
## 1 Month-to-month             0
## 2 One year                   0
## 3 Two year                   0
## 4 Month-to-month             1
## 5 Two year                   1
## 6 One year                   1
```

```
# Get the predicted probabilities

mult.pred <- augment(mult.model, newdata = mult.df, type.predict = "response")
mult.pred
```

```
## # A tibble: 6 x 4
##   Contract       SeniorCitizen .fitted .se.fit
##   <fct>                  <dbl>   <dbl>   <dbl>
## 1 Month-to-month             0  0.397  0.00869
## 2 One year                   0  0.104  0.00780
## 3 Two year                   0  0.0266 0.00380
## 4 Month-to-month             1  0.541  0.0165
## 5 Two year                   1  0.0466 0.00710
```

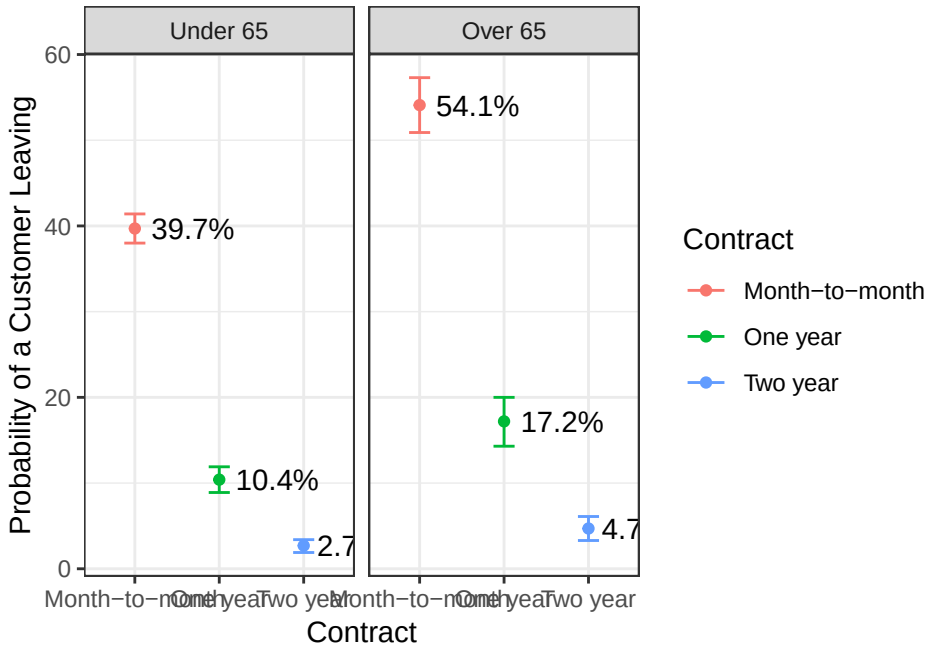
```
## 6 One year                   1  0.172  0.0145
```

```
# Quantify the uncertainty
mult.pred <- mult.pred %>%
  mutate(Pr_y = 100 * .fitted,
         lower.ci = 100 * (.fitted - (1.96 * .se.fit)),
         upper.ci = 100 * (.fitted + (1.96 * .se.fit))) %>%
  select(Contract, SeniorCitizen, Pr_y, lower.ci, upper.ci) %>%
  mutate_if(is.numeric, funs(round(., 1)))

mult.pred
```

```
## # A tibble: 6 x 5
##   Contract       SeniorCitizen  Pr_y lower.ci upper.ci
##   <fct>                  <dbl> <dbl>    <dbl>    <dbl>
## 1 Month-to-month             0  39.7     38       41.4
## 2 One year                   0  10.4      8.9     11.9
## 3 Two year                   0   2.7      1.9      3.4
## 4 Month-to-month             1  54.1     50.9     57.3
## 5 Two year                   1   4.7      3.3      6.1
## 6 One year                   1  17.2     14.3     20
```

Visualize it

```
sc.labels <- c("0" = "Under 65", "1" = "Over 65")

ggplot(data = mult.pred, aes(y = Pr_y, x = Contract, color = Contract)) +
  geom_point() +
  geom_errorbar(aes(ymin = lower.ci, ymax = upper.ci), width = 0.25) +
  geom_text(aes(label = paste0(Pr_y, "%")), color = "black",
            hjust = - .2) +
  labs(y = "Probability of a Customer Leaving") +
  facet_wrap(~ SeniorCitizen,
             labeller = labeller(SeniorCitizen = sc.labels)) +
  theme_bw()
```



Continuous predictor

```
tenure.model <- glm(Churn ~ MonthsTenure, data = churn.df, family = "binomial")
summary(tenure.model)
```

```
##
## Call:
## glm(formula = Churn ~ MonthsTenure, family = "binomial", data = churn.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1890  -0.8386  -0.4796   1.1823   2.3770
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.027313   0.042220   0.647    0.518
## MonthsTenure -0.038767   0.001405 -27.589   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8150.1  on 7042  degrees of freedom
## Residual deviance: 7191.9  on 7041  degrees of freedom
## AIC: 7195.9
##
## Number of Fisher Scoring iterations: 4
```

This is just the average, but the actual probabilities vary by level of the predictor

```
# Get the range of months of tenure
tenure.df <- tibble(MonthsTenure = seq(1, max(churn.df$MonthsTenure), 1))
tenure.df
```

```
## # A tibble: 72 x 1
##    MonthsTenure
##           <dbl>
##  1            1
##  2            2
##  3            3
##  4            4
##  5            5
##  6            6
##  7            7
##  8            8
##  9            9
## 10           10
## # ... with 62 more rows
```

```
# Calcutate the predicted probabilities
tenure.pred <- augment(tenure.model, newdata = tenure.df, type.predict = "response")
tenure.pred
```

```
## # A tibble: 72 x 3
##    MonthsTenure .fitted .se.fit
##           <dbl>   <dbl>   <dbl>
##  1            1   0.497 0.0103
##  2            2   0.487 0.0100
##  3            3   0.478 0.00979
##  4            4   0.468 0.00954
##  5            5   0.458 0.00929
##  6            6   0.449 0.00904
##  7            7   0.439 0.00879
##  8            8   0.430 0.00855
##  9            9   0.420 0.00831
## 10           10   0.411 0.00808
## # ... with 62 more rows
```

```
# Quantify the uncertainty
tenure.pred <- tenure.pred %>%
  mutate(Pr_y = 100 * .fitted,
         lower.ci = 100 * (.fitted - (1.96 * .se.fit)),
         upper.ci = 100 * (.fitted + (1.96 * .se.fit))) %>%
  select(MonthsTenure, Pr_y, lower.ci, upper.ci) %>%
  mutate_if(is.numeric, funs(round(., 1)))

tenure.pred
```

```
## # A tibble: 72 x 4
##    MonthsTenure  Pr_y lower.ci upper.ci
##           <dbl> <dbl>    <dbl>    <dbl>
##  1            1  49.7     47.7     51.7
##  2            2  48.7     46.8     50.7
##  3            3  47.8     45.9     49.7
##  4            4  46.8     44.9     48.7
```

```
##  5            5  45.8     44       47.7
##  6            6  44.9     43.1     46.7
##  7            7  43.9     42.2     45.7
##  8            8  43       41.3     44.7
##  9            9  42       40.4     43.7
## 10           10  41.1     39.5     42.7
## # ... with 62 more rows
```

```
# Visualization
ggplot(data = tenure.pred, aes(y = Pr_y, x = MonthsTenure)) +
  geom_line() +
  geom_ribbon(aes(ymin = lower.ci, ymax = upper.ci), alpha = .2) +
  labs(y = "Probability of a Customer Leaving",
       x = "Months of Customer Tenure") +
  theme_bw()
```